NEW TRACKING FILTER ALGORITHM USING INPUT ESTIMATION

THESIS

Corey M. Broussard, Captain, USAF
AFIT/GA/ENG/06-01

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/GA/ENG/06-01

NEW TRACKING FILTER ALGORITHM USING INPUT PARAMETER ESTIMATION

THESIS

Presented to the Faculty

Department of Aeronautics and Astronautics

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science (Astronautical Engineering)

Corey M. Broussard, BS

Captain, USAF

June 2006

AFIT/GA/ENG/06-01

NEW TRACKING FILTER ALGORITHM USING INPUT PARAMETER ESTIMATION

Corey M. Broussard, BS
Captain, USAF

Approved:

M. Pachter
Dr. Meir Pachter (Chairman) — 12 June 06 date

Peter S. Maybeck
Dr. Peter Maybeck (Member) — 12 June 06 date

Lt Col Juan Vasquez (Member) — 12 Jun 06 date

Abstract

A new method for the design of tracking filters for maneuvering targets, based on kinematic models and input signal estimation, is developed. The input signal's level u is considered a continuous variable and consequently the input estimation problem is posed as a purely parameter estimation problem. Moreover, the application of the new tracking filter algorithm is not contingent on distinguishing maneuvering and non-maneuvering targets, and does not require the detection of maneuver onset. The filter will automatically detect the onset of a maneuver. Furthermore, an estimate of the target's acceleration is also obtained with reasonable precision. This opens the door to the employment of advanced Augmented Proportional Navigation Missile guidance laws, which require an estimate of the target's state acceleration, recognizing the precision is not as good as that for position and velocity estimates. When the target dynamics and measurement equation are linear and input u is constant, then an unbiased estimate $\hat{u}$ of u and of the target's state is obtained, provided that an observability condition holds. It is shown that the critical observability condition holds for kinematic target motion models of interest.

Acknowledgments

First and foremost, I would like to thank my family for their unconditional support during my time at AFIT. Without their love and encouragement this endeavor would not have been possible. Thanks also to Dr. Pachter, for his guidance and support throughout the course of this thesis effort. I sincerely wish to work with you in the future.

Table of Contents

Page

List of Figures

Figure Page

List of Tables

Table Page

NEW TRACKING FILTER ALGORITHM
USING CONTINUOUS INPUT PARAMETER ESTIMATION

I. Introduction

The problem is posed of using a Kalman Filter (KF) to track a maneuvering target, that is, designing a tracking filter. The standard KF paradigm addresses the estimation of the state x of a dynamical system in "free fall" (no control) or of a controlled system when the input signal u is known. The standard KF arrangement is illustrated in Figure 1.

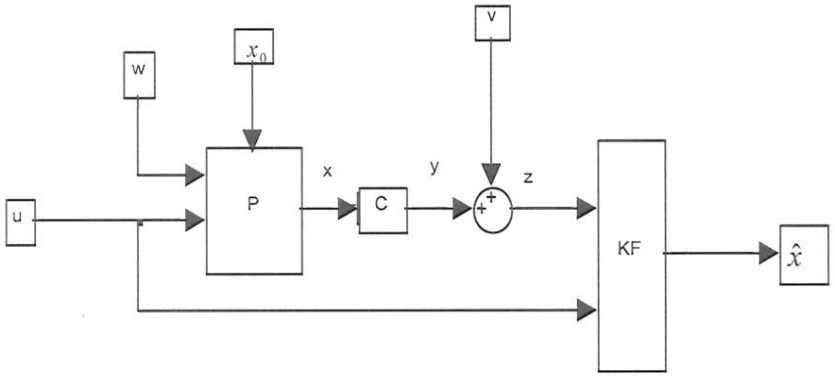


Figure 1: Standard Kalman Filter Arrangement. The state is x, the input signal is u, the output is y, w is process noise, the measurement noise is v and the state estimate produced by the KF is $\hat{x}$.

Hence, in the context of tracking, the KF paradigm is ideally suitable for tracking non-maneuvering targets for which the input signal $u(t) \equiv 0$. By "non-maneuvering", we mean no input/control from the pilot applied to the aircraft. In reality, one is most often faced with the problem of tracking maneuvering targets. Obviously, the target aircraft's

pilot's inputs u are not communicated to the tracking filter. Thus, the conventional KF paradigm is not always suitable for estimating the aircraft's state: in other words, the conventional KF paradigm is not suitable for designing tracking filters. To overcome this conceptual difficulty, starting with the pioneering work of Singer [6], the following tracking filter concepts have been proposed in the literature.

1. Tracking filters based on First-Order Gauss-Markov Acceleration (FOGMA) dynamics, for which the input signal driving the target's dynamics is stipulated to be a random signal, namely process noise [3].
2. Quantization of maneuver level and the use of Multiple Model Adaptive Estimation (MMAE) methods [4].
3. Maneuver detection-based schemes [1].
4. Multiple Hypotheses Trackers (MHT) [2].
5. Fixed gain Kalman Filters, referred to as $\alpha - \beta$ (i.e., "constant-velocity" modeling assumption) and $\alpha - \beta - \gamma$ (i.e., assuming the adequacy of "constant-acceleration" models of target dynamics) tracking filters [5].

In this thesis, the tracking of maneuvering targets is considered and the KF paradigm is extended. A novel method for the design of tracking filters for maneuvering targets based on input signal estimation is advanced. Estimation of u as well as x has been suggested before in tracking filter design [1]. These similar but not exact methodologies will be described in Chapter 2.

1.1 Problem Statement

We employ a basic concept from calculus for the design of tracking filters for maneuvering targets. The task is to track maneuvering/non-maneuvering targets adequately using only one Kalman Filter and attempt to estimate the pilot's inputs u using a moving/sliding window estimation algorithm, namely the unknown input signal is taken to be constant during a short time interval.

1.2 Approach

To address the concept outlined above, we take an approach rooted in calculus. Assume u(t) is a C^1 (continuously differentiable) function. According to Taylor's theorem, the value of u(t) at $t + \Delta T$ is

$$u(t+\Delta T) = u(t) + \left.\frac{du}{dt}\right|_t \Delta T + r(t;\Delta T) \ ,$$

and, for t fixed,

$$\lim_{\Delta t \to 0} \frac{r(t;\Delta T)}{\Delta T} = 0;$$

Indeed, according to Taylor's Theorem, the next term in the remainder is

$$r(t;\Delta T) = \frac{\Delta T^2}{2}\frac{d^2u}{dt^2}(t+\xi\Delta T),$$

where $0 \le \xi \le 1$ and the parameter ξ is not known.

Hence for this C^1 function, one goes on to use the approximation

$$u(t+\Delta T) \approx u(t) + \left.\frac{du}{dt}\right|_t \Delta T$$

provided ΔT is sufficiently small. This is tantamount to assuming

$$\frac{du}{d\tau}(\tau) = \left.\frac{du}{d\tau}\right|_{\tau=t} \equiv const \quad \forall \quad t \le \tau \le t + \Delta T$$

so that the linear/affine approximation holds

$$u(\tau) = u(t) + \left.\frac{du}{dt}\right|_{t} (\tau - t) \quad \forall \quad t \le \tau \le t + \Delta T$$

The residual $r(t;\Delta T)$ is the second derivative of u evaluated at σ, where $t \le \sigma \le t + \Delta T$ and $\sigma = t + \xi \Delta T$ is not known. Obviously, when u is linear/affine the above formula is exact. Similarly, if $u(t)$ is just continuous, the approximation is used

$$u(\tau) \approx u(t) \quad \forall \quad t \le \tau \le t + \Delta T,$$

for ΔT sufficiently (very) small.

Consider now a control system on the time interval $[0, \Delta T]$, ΔT small. The dynamics are

$$\frac{dx}{dt} = f[x(t), u(t)], \; x(0) = x_0, \quad 0 \le t \le \Delta T.$$

In the context of tracking filter design, the dynamics equations describe the target's dynamics. The kinematic variables with derivatives that are not featured in the kinematic equations are designated the control variables.

Assume the control signal $u(t)$ is continuous and set

$$u(\tau) \approx u(0) = u \quad \forall \quad 0 \le \tau \le \Delta T;$$

thus, the control signal is essentially constant (i.e., $\dot{u} = 0$) during the interval $[0, \Delta T]$.

Next, augment the state vector with the control vector, that is, treat the control vector as a state variable and use the augmented state $X \equiv \begin{pmatrix} x \\ u \end{pmatrix}$. One then obtains the augmented dynamical system

$$\frac{dX}{dt} = \begin{bmatrix} f(x,u) \\ 0 \end{bmatrix}, \quad X_0 = \begin{bmatrix} x(0) \\ u(0) \end{bmatrix} = \begin{bmatrix} x_0 \\ u \end{bmatrix}.$$

One has now obtained an augmented dynamical system in "free fall". In other words, the model of the system is a homogenous differential equation. This, in turn, allows us to use the standard KF paradigm for (augmented) state estimation. However, since the control signal components of the augmented state are not known, a tracking filter implementation requires one to:

1. Keep the estimation interval ΔT short, so that $u \equiv const$ is a valid approximation of reality. Thus, we'll choose the length of the estimation interval commensurate with the bandwidth of the input signal $u(t)$.
2. A moving/sliding window approach is chosen to allow for a possible change/jump in the control signal. One could just change $\dot{u} = 0$ to $\dot{u} = 0 + w$ (i.e, add pseudonoise to the model) and just use a conventional Kalman-like filter on the augmented X with no moving window approach; however, this would mitigate flexibility in the algorithm to "adapt" to harsh maneuvers.
3. When one moves on to a new estimation window, we restart the estimation process. Thus, in the new estimation window one is allowed to use prior information on the continuous x components of the augmented state, but not on

the u components of the augmented state – this, according to our stipulation that the control signal is not known. Thus, the control signal's estimate $\hat{u}$ will be data-driven. We pose a least squares best estimate of a constant u over ΔT to fit the data. If u takes on a step change, we recognize the model is invalid and hence, provide the user the detection of a new maneuver.

In other words,

1. The estimation interval length ΔT is small. Specifically, it is chosen commensurate with the known input signal's bandwidth BW,

$$\Delta T \leq \frac{1}{2BW}. \qquad (1)$$

2. A moving/sliding window estimation process is used. As one moves from window to window, prior information on the x state component is used, since $x(t)$ is continuous. No prior information on u is used.

3. We use a smoothing algorithm [4] over each ΔT to get a good estimate of the initial $\hat{x}$ value for the next window in order to take advantage of data collected from the previous estimation window. Thus, the state estimate and the state estimation error covariance are smoothed for use in the new estimation window, whereas the initial uncertainty about the control signal u is set high. In the new estimation window, the control signal's initial covariance is re-set large to make the point that in the new estimation window the control signal's level is NOT known.

1.3 Chapter Summary

This research advances the concept of input estimation for designing tracking filters for maneuvering targets. This section summarizes the layout of the thesis. Chapter II provides the background of input estimation. Chapter III discusses the methodology of the moving window with smoothing algorithm. Chapter IV presents the simulation test results. Chapter V discusses the results and their relevance to the current and future uses of tracking. Chapter VI summarizes the research presented, and provides recommendations for the further development of this new tracking algorithm.

II. Background

2.1 Bar-Shalom's Input Estimation Method

Bar-Shalom [1] has considered input estimation along with state estimation in tracking filter design. In both his and our work, the input signal is considered constant during the estimation interval ΔT. Most importantly, the input estimates rendered by both approaches are similar. That's also where the commonality of the approaches ends. In our work, the input u is considered a continuous (not discrete) variable. As a result, Input Estimation (IE) is posed as a purely parameter estimation problem. Of course, the estimated input could end up being $\equiv 0$, in which case one refers to a non-maneuvering target; or, it could transpire that $\hat{u} \neq 0$. The point is that no damage is done by allowing u to be a continuous parameter, even when the target is not maneuvering and the true input level $u \equiv 0$.

Our algorithm, and also Bar-Shalom's algorithm, yields an unbiased estimate of u, namely $\hat{u}$. Both algorithms yield the same answer in this linear estimation problem. To summarize our approach: we recognize that IE is a continuous parameter estimation problem. Hence, our approach is direct and the estimation algorithm is straightforward. Concerning the alternative Bar-Shalom approach, we argue that it is not necessary to draw a distinction between a maneuvering and non-maneuvering target and by doing so to be drawn into discrete decision problems in which one, preliminary to parameter estimation, must address the detection problem and test the maneuvering/non-maneuvering hypotheses. There is no need to address a detection problem and decide whether the tracked target is maneuvering or is not maneuvering, and if it is maneuvering, then estimate the maneuver level u. Bar-Shalom's approach and ensuing estimation

algorithm is more complex. As previously mentioned, both solutions are the same in this linear estimation problem.

In summary, our approach treats the input u as a continuous parameter as opposed to a discrete parameter and we do not require a separate algorithm for maneuver detection.

III. Methodology

To understand the methodology of the algorithm used in this thesis fully, we explore the concept using a linear estimation example. We then describe the moving window algorithm and smoothing algorithm used in our tracking filter.

3.1 Linear Estimation Example

The herein discussed estimation concept is now illustrated in the linear world. The linear control system

$$\dot{x} = Ax + Bu \qquad x(0) = x_0 \qquad 0 \le t \tag{2}$$

$$y = Cx$$

is considered. The control signal u is not available to the estimator. It is desired to estimate the control system's state. To this end, the input u will be estimated.

A piecewise constant approximation of the control signal u is considered. Specifically, during each estimation interval the control signal u is assumed constant. Thus, take $u \equiv const$ during the initial short estimation time interval $0 \le t \le \Delta T$ and consider the augmented dynamical system

$$\frac{d}{dt}\begin{pmatrix} x \\ u \end{pmatrix} = \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix}\begin{pmatrix} x \\ u \end{pmatrix} \quad , x(0) = x_0 \ , \ u(0) = u \ , \ u \text{ not known} \tag{3}$$

The augmented measurement equation now is

$$y = \begin{bmatrix} C & 0 \end{bmatrix}\begin{pmatrix} x \\ u \end{pmatrix}, \ 0 \le t \le \Delta T \, .$$

In a deterministic setting, one is thus motivated to pose the observability question: can one determine the initial state x_0 and the unknown parameter u (which is now a

component of the augmented state vector $X \equiv \begin{pmatrix} x \\ u \end{pmatrix}$), given the observation record $y(t)$, $0 \le t \le \Delta T$?

The question is posed of whether this new augmented dynamical system is observable. Thus, suppose $x, x_0 \in R^n$, $u \in R^m$, and $y \in R^l$. One must calculate the observability matrix

$$O = \begin{bmatrix} C^T & A^T C^T & \left(A^T\right)^2 C^T & \cdots & \left(A^T\right)^{n+m-1} C^T \\ 0 & B^T C^T & B^T A^T C^T & \cdots & B^T \left(A^T\right)^{n+m-2} C^T \end{bmatrix}_{(n+m)\times l(n+m)} \tag{4}$$

and one is naturally interested in $rank(O)$. The question is posed:

$$rank(O) \overset{?}{=} n+m$$

where n is the dimension of $x(t)$ and m is the dimension of $u(t)$.

If the answer is on the affirmative, the augmented system is observable and the state components, including the constant control signal u, can be calculated from the measurement record $y(t), 0 \le t \le \Delta T$. As far as observability is concerned, namely, in the deterministic case, the estimation interval ΔT can be arbitrarily small. By moving the estimation window to $[kdt, kdt + \Delta T]$, $k = 1, 2, \ldots$, an approximation of the continuous control signal $u(t)$, $0 \le t$, is obtained. In general, we will show the following holds.

THEOREM 1 In a stochastic setting and when a Kalman Filter is used for state estimation, a data driven complete state estimate can be obtained, including the control/maneuver level parameter u, without prior state (including u) information, that is,

one is then able to initialize the KF with $P_0 = \alpha I_m$, $\alpha \to \infty$, iff the augmented dynamical system is observable.

□

The observabilty condition is not overly restrictive. Indeed, consider the target kinematics

$$\dot{x} = u$$

and the measurement equation

$$y = x$$

We set $u \equiv const$ that is, $\dot{u} = 0$ and obtain the augmented system

$$\frac{d}{dt}\begin{pmatrix} x \\ u \end{pmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{pmatrix} x \\ u \end{pmatrix}, \quad \begin{pmatrix} x \\ u \end{pmatrix}(0) = \begin{pmatrix} x_0 \\ u_o \end{pmatrix}, \quad t \geq 0$$

This system is observable. The point is that this system is observable also when the state $x \in R^2$, in which case one would be modeling the situation in which an aircraft is flying at a constant altitude and is being tracked by a two-dimensional radar (results of this model will be shown in Chapter 4).

Consider the linear control system

$$\frac{d}{dt}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \qquad \begin{matrix} x_1(0) = x_{1_0} \\ x_2(0) = x_{2_0} \end{matrix}$$

The measurement equation is

$$y = x_1$$

The augmented state vector is

$$X = \begin{pmatrix} x_1 \\ x_2 \\ u \end{pmatrix}$$

and state space representation of the augmented dynamical system is

$$\frac{d}{dt}\begin{pmatrix} x_1 \\ x_2 \\ u \end{pmatrix} = \left[\begin{array}{ccc} 0 & 1 & 0 \\ 0 & 0 & 1 \\ \hline 0 & 0 & 0 \end{array}\right]\begin{pmatrix} x_1 \\ x_2 \\ u \end{pmatrix} \quad , \quad y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}\begin{pmatrix} x_1 \\ x_2 \\ u \end{pmatrix}$$

We calculate the observability matrix

$$O = \left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \hline 0 & 0 & 1 \end{array}\right]$$

and $rank(O) = 3$. Hence, the system is observable and one can calculate the initial state and the parameter u from a measurement record $y(t)$, $0 \le t \le \Delta T$. Although this holds true for arbitrarily small $\Delta T \ge 0$, the presence of measurement noise motivates us to use the "longest" possible estimation interval ΔT; the latter is circumscribed by the input's dynamics, that is, bandwidth BW [Hz] of the input signal u and is set according to eq. (1), namely

$$\Delta T \le \frac{1}{2BW} \quad [\text{sec}]$$

Thus, a novel tracking filter synthesis method for maneuvering targets based on input estimation and kinematic modeling is developed. In the kinematic model, the fastest variable, that is the variable with the highest derivative, is designated the control variable. For example, consider an aircraft flying at a constant altitude. A good kinematic model is obtained as follows: the aircraft's speed and course angle are declared the control variables, and by setting the latter constant during the short estimation interval (≡ setting their derivative equal to 0), a parameter estimation problem ensues. Our tracking filter synthesis method calls for augmenting the system's state with the control

variables and, in the process, one obtains a dynamical system "in free fall" which is amenable to conventional state estimation using Kalman filtering. Indeed, Kalman filtering is about state estimation. The control variable u, which augments the control system's state and quantifies/models the maneuver's level, is not known. Therefore, in our input estimation-based tracking filter design methodology, the u estimate does not rely upon the availability of prior information about it and is exclusively data driven. That's why our tracking filter design method is contingent on the augmented dynamical system being observable.

3.2 Moving Window Algorithm

A moving window algorithm to allow for a time-varying input signal is used. The algorithm is first demonstrated symbolically. The prepared-ahead-of-time data (measurements): $z_1, \ldots, z_{M+N-1}$

$$z_k = y_k + v_k \qquad k = 1, \ldots, M+N-1; \qquad v_k \cong N\left(0, \sigma^2\right)$$

where M is the number of data windows and N is the final time. The moving window KF algorithm will be exercised M times. Consider window # l, $l = 1, \ldots, M$ where the KF operates on measurements $z_1, \ldots, z_{l+N-1}$.

The continuous-time dynamics are

$$\frac{d}{dt}X = A_a X \ , \ X(0) = X_0 \ , \ 0 \le t$$

where

$$X = \begin{pmatrix} x_1 \\ x_2 \\ u \end{pmatrix} \quad \text{and} \quad A_a = \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

The discrete-time dynamics are

$$X_k = \begin{bmatrix} 1 & dt & \frac{1}{2}dt^2 \\ 0 & 1 & dt \\ 0 & 0 & 1 \end{bmatrix} X_{k-1} \qquad k = 1, \ldots, l+N-1;$$

The measurement equation is

$$z_k = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} X_k + v_k \qquad , \; v_k \cong N\left(0, \sigma^2\right)$$

The KF is initialized with X_l^-, P_l^-

For $l = 1$ use

$$X_1^- = X_0 = \begin{pmatrix} x_{1_0} \\ x_{2_0} \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$P_1^- = \begin{bmatrix} \sigma_1^2 & 0 & 0 \\ 0 & \sigma_2^2 & 0 \\ 0 & 0 & \infty \end{bmatrix}$$

We used true $u(t) = 1$ as the initial condition of the input in most of the experiments. Even though this may give the filter artificial knowledge of truth to get good initial transient, the input signal's initial condition is not that important since its covariance is set high at the beginning of each window. Rather than make $[P_1^-]_{33}$ huge, it might be better to use an inverse covariance form of Kalman filter and let $[(P_1^-)^{-1}]_{33}$ be zero instead, in order to avoid numerical precision problems that can occur with a huge $[P_1^-]_{33}$; this, however, was not exercised in this research.

Step through the KF algorithm and obtain

$$\hat{X}_l^+, \ldots, \hat{X}_{l+N-1}^+$$

$$P_l^+, \ldots, P_{l+N-1}^+$$

Set

$$u(k) = \left(\hat{X}_{l+N-1}\right)_3 \qquad \forall \qquad k = l-1, \ldots, l+N-2$$

$$\sigma_u(k) = \sqrt{\left(P_{l+N-1}^+\right)_{3,3}} \qquad \forall \qquad k = l-1, \ldots, l+N-2$$

We then implement a smoothing algorithm and obtain $\hat{X}_l^{(s)}, P_l^{(s)}$ to provide the initial conditions for the state values at the next window by setting

$$\hat{X}_{l+1}^- = \hat{X}_l^{(s)} \text{ and } P_{l+1}^- = \begin{bmatrix} \left(P_l^{(s)}\right)_{2\times 2} & \begin{matrix} 0 \\ 0 \end{matrix} \\ \begin{matrix} 0 & 0 \end{matrix} & \infty \end{bmatrix}$$

This is done to take advantage of data already collected from the previous window. We use a fixed-point smoothing algorithm [4] since there is certain point in time at which the value of the system state is desired. The initial conditions of the next window are one *dt* (on the order of a tenth of a second in most cases) from the initial conditions of the previous window; therefore, the use of a smoother aids the algorithm in obtaining better state estimates.

3.3 Fixed-Point Smoothing Algorithm

As mentioned above, a fixed-point smoothing algorithm is used to obtain the initial conditions for each incremental window. From [4], starting from initial condition

$$\hat{x}(t_i \mid t_i) = \hat{x}(t_i^+)$$

with $\hat{x}(t_i^+)$ obtained from a concurrently running Kalman filter, the relation

$$\hat{x}(t_i \mid t_j) = \hat{x}(t_i \mid t_{j-1}) + W(t_j)\left[\hat{x}(t_j^+) - \hat{x}(t_j^-)\right]$$

is solved for fixed t_i , letting $t_j = t_{i+1}, t_{i+2}, \ldots, t_f$, with the gain matrix $W(t_j)$ evaluated recursively as

$$W(t_j) \stackrel{\Delta}{=} \prod_{k=i}^{j-1} A(t_k) = W(t_{j-1}) A(t_{j-1})$$

$$A(t_k) \stackrel{\Delta}{=} P(t_k^+) \Phi^T(t_{k+1}, t_k) P^{-1}(t_{k+1}^-)$$

The values for $\hat{x}(t_j^+)$ and $\hat{x}(t_j^-)$ in this expression are obtained from the Kalman filter iterating forward in time. In fact,

$$[\hat{x}(t_j^+) - \hat{x}(t_j^-)] = K(t_j)[z_j - H(t_j)\hat{x}(t_j^-)]$$

The error committed by this smoother is Gaussian and zero mean for $j = i$, $i + 1, \ldots, f$, with covariance:

$$P(t_i \mid t_j) = P(t_i \mid t_{j-1}) + W(t_j)[P(t_j^+) - P(t_j^-)]W^T(t_j)$$

solved forward for $t_j = t_{i+1}, t_{i+2}, \ldots, t_f$ from the initial condition

$$P(t_i / t_i) = P(t_i^+)$$

3.4 Summary

Summarizing, the moving window and fixed-point algorithms together make up this new tracking filter paradigm. The moving window algorithm is used to allow for any sudden maneuvers (step changes) of the aircraft and the smoothing algorithm is used to take advantage of measurements from the previous window. The initial conditions of the state estimates (not the control signal) for each subsequent window are provided from the smoothing algorithm performed on the previous window.

IV. Simulations and Results

4.1 Moving Window Simulations

Controlled experiments were conducted using the system dynamics and input control signals from a variety of models. The level of modeling of target kinematics has been gradually increased, starting out from constant acceleration models, to sinusoidal functions, to more realistic aircraft/rocket models. Each model will be explained and results presented. Synthetic measurements were generated by corrupting them with white Gaussian noise. We show that the estimation algorithm properly produces good estimates of the control parameter most of the time and, in addition, estimates fall inside a band of +/- one predicted σ, also provided by the algorithm, around the true parameter. For a well tuned filter based on a good model, 68% of the estimates should fall within +/- 1 σ pf true; 95% should fall within +/- 2 σ of true. The software used was MATLAB. Monte Carlo runs were generated of the various simulations. After each test concluded, the results were analyzed and plotted using MATLAB.

4.1.1 Calibration Experiment

A calibration experiment was conducted first to determine the relationship between the Signal-to-Noise Ratio (SNR) of the measurements to the non-dimensional number of sample periods in each estimation window, β. We vary the SNR of the data and the discretization step of the Kalman Filter of one window. We allow the window to "expand" until certain tolerances are met. The following kinematic model was used:

$$\ddot{x} = u \ , \ x(0) = x_0 \ , \ \dot{x}(0) = \dot{x}_0 \ , \ u(t) = 1(t) \ , \ 0 \le t$$

$$y = x$$

This yields the augmented third-order kinematics:

$$\ddot{x} = u \qquad x(0) = x_0 \qquad \dot{x}(0) = \dot{x}_0$$

$$\dot{u} = 0 \qquad u(0) = u_0 \qquad 0 \le t$$

$$y = x$$

The truth signal in this simulation experiment is

$$x(t) = \frac{1}{2}t^2$$
$$\dot{x}(t) = t$$
$$u(t) = 1$$

Let the augmented state be

$$X = \begin{pmatrix} x \\ \dot{x} \\ u \end{pmatrix}$$

so that

$$\dot{X} = A_C X \qquad X(0) = X_0 = \begin{pmatrix} x_0 \\ \dot{x}_0 \\ u_0 \end{pmatrix} \qquad 0 \le t$$

$$y = C_C X$$

where

$$A_C = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \qquad C_C = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}$$

The discrete-time model of the system is

$$X_{k+1} = A_D X_k \qquad X_0 \equiv X_0 \qquad k = 0, 1, \ldots$$

$$y_{k+1} = C_D X_{k+1}$$

where

$$X_k = X(k\,dt)\,,\ y_k = y(k\,dt)\,,\ k = 1, 2, \ldots;$$

and

$$A_D = \begin{bmatrix} 1 & dt & \frac{1}{2}dt^2 \\ 0 & 1 & dt \\ 0 & 0 & 1 \end{bmatrix} \qquad C_D = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}$$

The measurement equation for the expanding window Kalman Filter

$$z_k = y_k + v_k \qquad v_k \cong N\left(0, \sigma_v^2\right) \qquad k = 1, 2, \ldots, N$$

and the measurement noise standard deviation σ_v used in the simulation is quantified by the Signal-to-Noise Ratio (SNR) parameter

$$SNR = \frac{Amp}{\sigma_v} = \frac{1}{\sigma_v} \quad \Rightarrow \quad \sigma_v = \frac{1}{SNR}$$

The expanding window Kalman Filter was initialized using

$$X_0 = \begin{pmatrix} x_0 \\ \dot{x}_0 \\ u_0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

with covariance

$$P_0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 10^6 \end{bmatrix}$$

The initial covariance of both dynamic states (not the control level u) are set to zero to simulate that they will be "perfectly known" at the beginning of each window. Since the initial conditions of each incremental window are the smoothed estimates from the

previous window, this assumption is adequate for our purpose of testing window length. Again, the input u's variance is set large to emphasize it is not known at window onset. We define the control level error tolerance as the accuracy of the estimated control signal. For example, a control level error tolerance of 0.01 is

$$|u - \hat{u}| \leq 0.01$$

Results of the calibration simulations are summarized in Tables 1, 2, and 3 for control level error tolerances 1, 0.1 and 0.01 respectively. Each entry in the table is the number of sample periods required to achieve the desired tolerance level for the given SNR and *dt*. The tables provide the length of the estimation interval for a given *dt* and SNR to achieve the desired control level error tolerance of one Monte-Carlo run. So, for a *dt* of 0.1 and an SNR of 10, the length of the estimation interval needs to be 9 sample periods in order to come within 1 unit of the true control signal. The mean error ($|u - \hat{u}|$) averaged over all Monte Carlo runs was considered but proved difficult to analyze since error is "accumulated" over each run. Again, this test was conducted simply to get an idea of the number of sample periods in each estimation window and could be further tuned during each simulation.

Table 1: Control Level Error Tolerance ≡ 1

dt / SNR	0.3	0.2	0.1	0.04	0.02	0.01	0.005	0.002	0.001
1	4	7	10	21	46	84	101	258	471
5	4	6	10	15	36	77	82	195	387
10	4	6	9	12	25	66	93	181	315
20	4	3	5	11	22	35	84	164	265
50	3	4	6	13	18	43	80	115	222
100	3	3	5	13	24	29	45	105	197

Table 2: Control Level Error Tolerance ≡ 0.1

dt / SNR	0.3	0.2	0.1	0.04	0.02	0.01	0.005	0.002	0.001
1	12	17	22	51	148	241	222	592	989
5	6	10	22	36	74	145	227	515	867
10	8	13	19	31	72	107	187	514	910
20	4	12	20	33	50	103	198	376	512
50	5	9	16	25	38	82	134	315	462
100	5	9	7	38	37	90	150	329	388

Table 3: Control Level Error Tolerance ≡ 0.01

dt / SNR	0.3	0.2	0.1	0.04	0.02	0.01	0.005	0.002	0.001
1	25	44	84	208	322	538	783	1002	1002
5	27	28	52	134	169	265	677	954	1002
10	18	32	56	99	138	323	511	1001	1002
20	14	24	47	48	130	257	452	693	1002
50	13	24	35	61	160	182	441	872	967
100	13	14	30	58	78	226	273	566	782

These tables provide good insight into the number of sample periods required within the estimation interval. For a given SNR, the smaller the *dt*, the more sample periods required in the estimation interval. Likewise, for a given *dt*, the higher the SNR, the fewer sample periods required in the estimation interval. Finally, the higher the control level error tolerance, the more sample periods required in the estimation interval. The goal is to keep the estimation interval as short as possible but not at the expense of estimate accuracy. An estimation interval of 30 was used in the following simulations.

4.1.2 Constant Acceleration Model

The moving/sliding window algorithm (MWA) was tested using the same constant acceleration model. The purpose of this model was to verify the smoothing algorithm and use the results as a basis for future simulations. The results are presented in Figure 2.

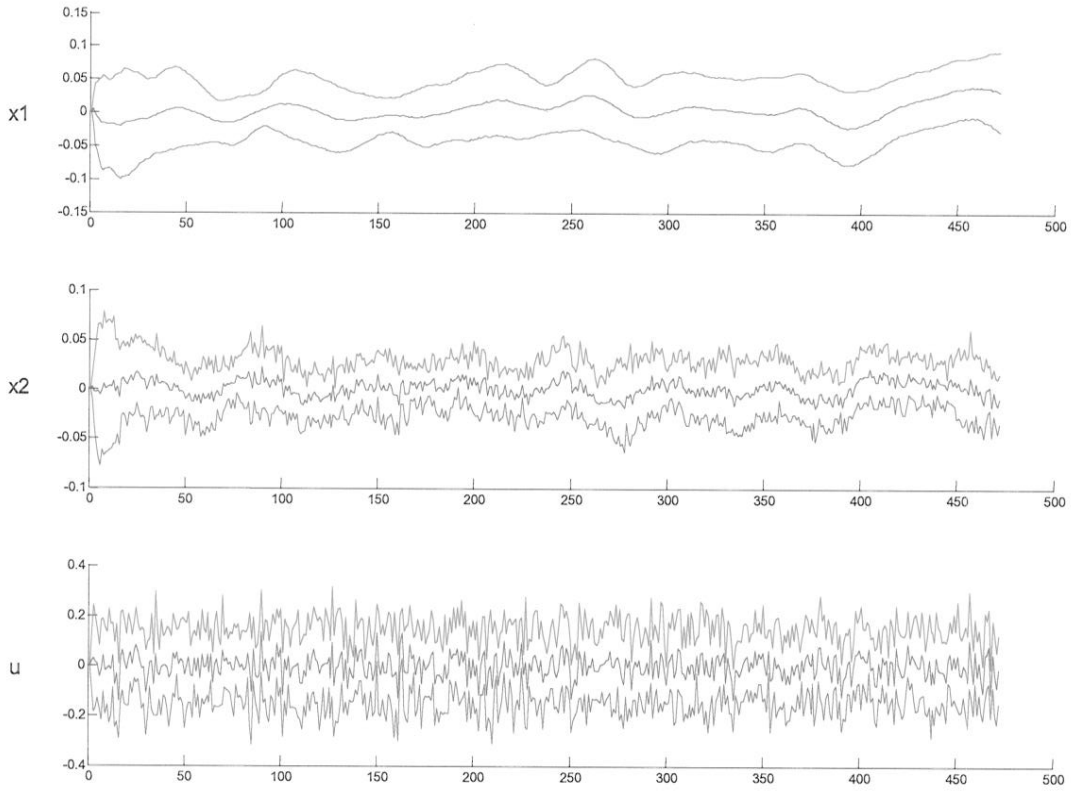


Figure 2: Ten Monte Carlo run analysis of the Constant Acceleration kinematic model using a Moving Window: $\sigma_k = 0.1$, $\beta = 30$, $dt = 0.1$. We show the True Mean Error +/- One Sigma (Monte Carlo evaluated).

This tested the efficiency of the smoothing algorithm using a constant acceleration input of 1. RMS errors are relatively small and the input signal, u, the acceleration, is being tracked with great precision. Consequently, the state components are properly tracked.

4.1.3 One Dimensional Sinusoidal Model

The input estimation-based MWA algorithm was tested using a sinusoidal control signal $u(t) = \cos(2\pi ft)$. The analytic solution is

$$x_1(t) = \frac{1}{4\pi^2 f^2}[1 - \cos(2\pi ft)]$$

$$x_2(t) = \frac{1}{2\pi f}\sin(2\pi ft)$$

Without loss of generality, we chose $f = 1$ *Hz*. We will also consider $f = 0$, in which case a constant unit step input is considered.

Let the estimation interval

$$\Delta T = \frac{1}{f_s}$$

Thus the length of the estimation interval is quantified by the f_s ([Hz]) parameter. One wishes to find an adequate moving window estimation interval of length ΔT, from which the input u is accurately estimated. Thus, set

$$\frac{f_s}{f} = \alpha \quad , \quad \alpha = 2, 3, 4, 5, 8, 10.$$

α is set higher than the Nyquist rate of 2 so that the bandwidth constraint is satisfied. The additional sampling rate f_{ss} sets the discretization time step dt in the discrete-time Kalman filter used herein. Since the dimension of the parameter space is 3 ($X = (x_1, x_2, u)$), set $\beta \equiv$ length of discrete-time estimation interval of the Kalman Filter, $\beta \geq 3$. Thus

$$\beta = \frac{f_{ss}}{f_s}, \quad \beta = 3, 5, 9, 20, 50.$$

Recall the measurement noise standard deviation σ used is quantified by the Signal-to-Noise Ratio (SNR) parameter

$$SNR = \frac{Amp}{\sigma} = \frac{1}{4\pi^2 f^2 \sigma} \quad \Rightarrow \quad \sigma = \frac{1}{4\pi^2 f^2 SNR}$$

We experimented with SNR = 1 (0 dB), 5, 10, 100, 200, 500, 1000 (60 dB).

First, a constant $u(t) \equiv 1$ calibration experiment was conducted to obtain the appropriate number of sample periods in a window. Depending on the SNR, the constant u experiment gave us an idea of how long the moving estimation window should be in order to obtain good state and input estimates. The truth maneuvering target's trajectory is, as before,

$$x_1(t) = \frac{1}{4\pi^2 f^2}\left(1 - \cos(2\pi f t)\right)$$

The results are presented in Figure 3. We will also compare the performance of our filter to an MMAE based tracking filter in Chapter 4.

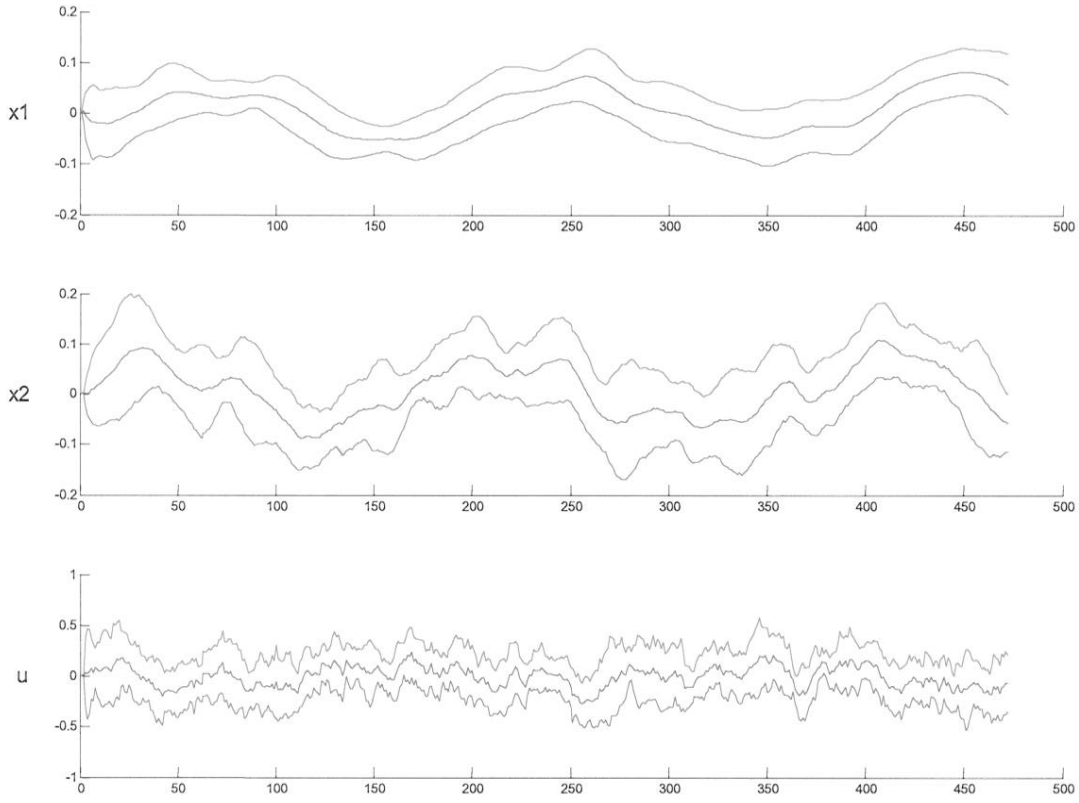


Figure 3: Ten Monte Carlo run analysis of the Sinusoidal Function simulation: $\alpha = 6.3$, $\beta = 30$, $f = 0.16$, $f_s = 1.0027$, $f_{ss} = 30$, $dt = 0.03$, $dT = 0.997$, $\sigma_k = 0.1$. We show the Mean Error +/- One Sigma (Monte Carlo evaluated).

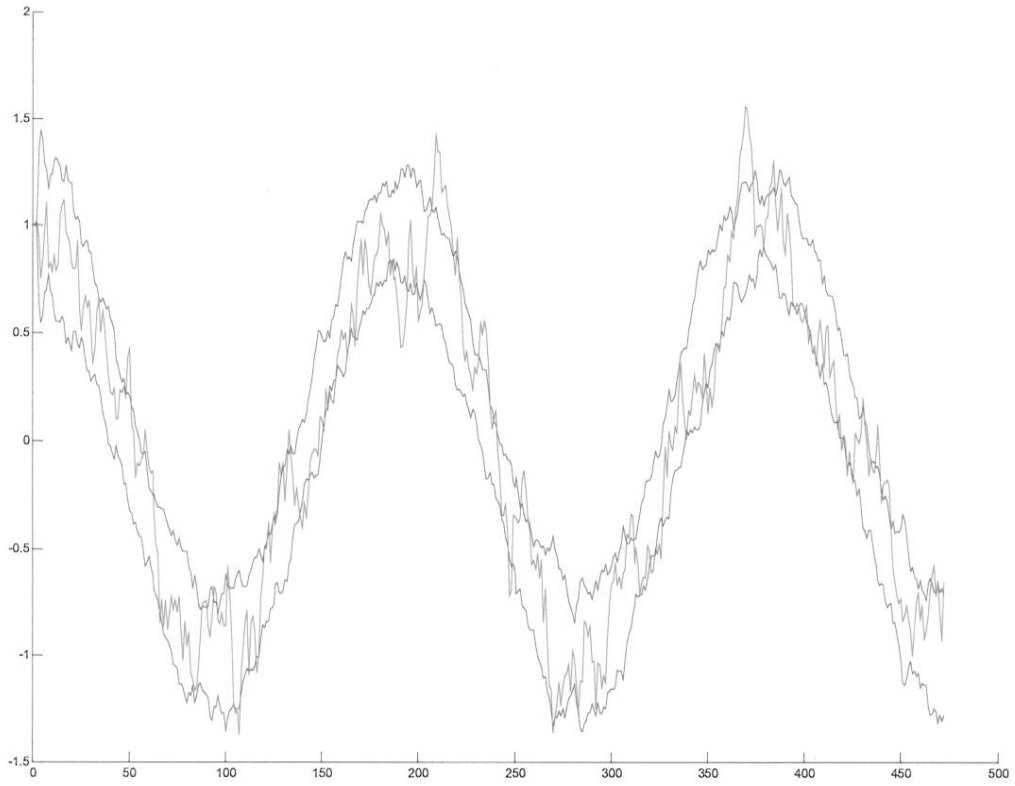


Figure 4: Estimated Input (red) and [True Input +/- One Sigma Error](Monte Carlo evaluated) (blue). The same parameters used in Figure 3 are illustrated. The estimated input is from one Monte Carlo run.

Figures 3 and 4 illustrate how well the MWA performs against a constant acceleration of 1. The estimation interval of length 30 is successful in producing adequate filter performance. The moving window algorithm tracks the sinusoidal control signal very well; it remains within the band of +/- one sigma (Monte Carlo evaluated) around true parameter. Even though this plot indicates rather "conservative" tuning, it provides a starting point (an estimation interval of length 30) for future simulations.

4.1.4 Aircraft Kinematic Model

We now test the algorithm against a more realistic model of aircraft motion. The aircraft's motion is confined to the Euclidean plane. The control variables are target course angle H and speed V. We observe the targets position (x, y), as when radar measurements are taken and $x = R\cos(\gamma)$, $y = R\sin(\gamma)$, where R and γ are the target's range and bearing, respectively.

$$\begin{aligned} \dot{x} &= V\cos(H) & x(0) &= x_0 \\ \dot{y} &= V\sin(H) & y(0) &= y_0 \end{aligned} \qquad 0 \le t$$

Thus

$$\begin{aligned} \dot{x} &= u_1 & x(0) &= x_0 \\ \dot{y} &= u_2 & y(0) &= y_0 \end{aligned} \qquad 0 \le t$$

During a measurement interval window, the control variables u_1 $(= V\cos(H))$ and u_2 $(= V\sin(H))$ are taken to be essentially constant. This should provide an adequate approximation if the estimation interval is small.

Therefore, the augmented dynamics in $\Re^4$ are

$$\begin{aligned} \dot{X} &= U & X(0) &= X_0 \\ \dot{U} &= 0 & U(0) &= 0 \end{aligned} \qquad 0 \le t$$

where

$$X \equiv \begin{pmatrix} x \\ y \end{pmatrix} \in \Re^2 \qquad U \equiv \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \in \Re^2$$

In matrix notation, the augmented dynamics in $\Re^4$ are

$$\frac{d}{dt}\begin{pmatrix} X \\ U \end{pmatrix} = \begin{bmatrix} 0_{2\times 2} & I_2 \\ 0 & 0 \end{bmatrix}\begin{pmatrix} X \\ U \end{pmatrix}$$

and the measurement

$$Y = \begin{pmatrix} x \\ y \end{pmatrix}$$

so that the measurement equation is

$$Y = \begin{bmatrix} I_2 & 0_{2\times 2} \end{bmatrix} \begin{pmatrix} X \\ U \end{pmatrix}$$

Thus

$$A_{4\times 4} = \begin{bmatrix} 0_{2\times 2} & I_2 \\ 0_{2\times 2} & 0_{2\times 2} \end{bmatrix} \qquad C_{2\times 4} = \begin{bmatrix} I_2 & 0_{2\times 2} \end{bmatrix}$$

The pair (A, C) is observable

Therefore, according to Theorem 1, one can obtain data-driven estimates of the input U and there is no need to use prior information. Indeed, observability implies that, from the measurement record $[x(t), y(t)]$, $0 \le t \le \Delta T$, one can obtain data-driven estimates of the parameter (u_1, u_2) without the use of prior information on the parameter. We do not want to use prior information on the parameter which are control variables and at the pilot's discretion.

Once the control variables u_1 and u_2 have been estimated, one obtains an estimate of the aircraft's velocity and the course angle (to first order)

$$\hat{V} \approx \sqrt{\hat{u}_1^2 + \hat{u}_2^2}$$

$$\hat{H} \approx \arctan\left(\frac{\hat{u}_2}{\hat{u}_1}\right)$$

Finally,

$$\hat{x}(\Delta T) = \hat{x}_0 + \hat{u}_1 \Delta T$$
$$\hat{y}(\Delta T) = \hat{y}_0 + \hat{u}_2 \Delta T$$

No smoothing is required when the parameters (u_1, u_2) is obtained by solving a linear regression (in batch).

For the aircraft kinematic model simulation, the dynamics model used was

$$\begin{aligned} \dot{x} &= V\cos(H) & x(0) &= 0 \\ \dot{y} &= V\sin(H) & y(0) &= 0 \\ H &= \Omega t & H(0) &= 0 \end{aligned} \qquad 0 \le t \le \frac{2\pi}{\Omega}$$

and therefore for constant Ω, the truth model becomes

$$x(t) = \left(\frac{V}{\Omega}\right)\sin(\Omega t) + x_0$$
$$y(t) = \left(\frac{V}{\Omega}\right)[1 - \cos(\Omega t)]$$

Table 4: Parameters for Aircraft Kinematic Model Simulation

$V_{true} = 200$ m/s	$\alpha = 125$	$f = 0.016\ Hz$	$dT = 0.5$ sec
$\sigma_k = 40.7$ m	$\beta = 30$	$f_s = 1.95 Hz$	$dt = 0.017$ sec
$\Omega = 0.0982$ rad/sec	$n = 3$	$f_{ss} = 58.61 Hz$	$x_0 = 20000$ m

This table summarizes the parameters of interest in the aircraft kinematic model. This simulation presented a more realistic application of the moving/sliding window and also tested it against considerably high measurement noise intensity levels and strong target maneuvers.

The tracking/estimation results for the maneuvering aircraft simulation are presented in Figures 5-7.

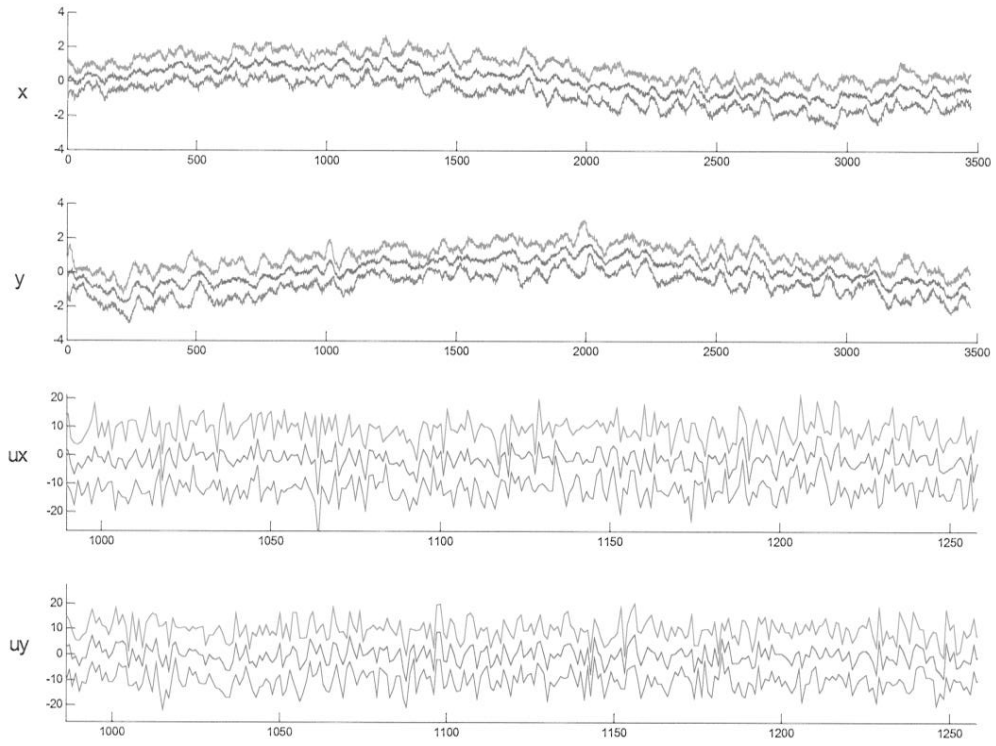


Figure 5: Mean Error +/- One Sigma (x, y, u_1, u_2) (Monte Carlo evaluated) (10 Monte Carlo Runs)

The RMS error of the control signal is not as good as in previous simulations. This is expected due to high measurement noise (low SNR). Good tracking performance is achieved nonetheless.

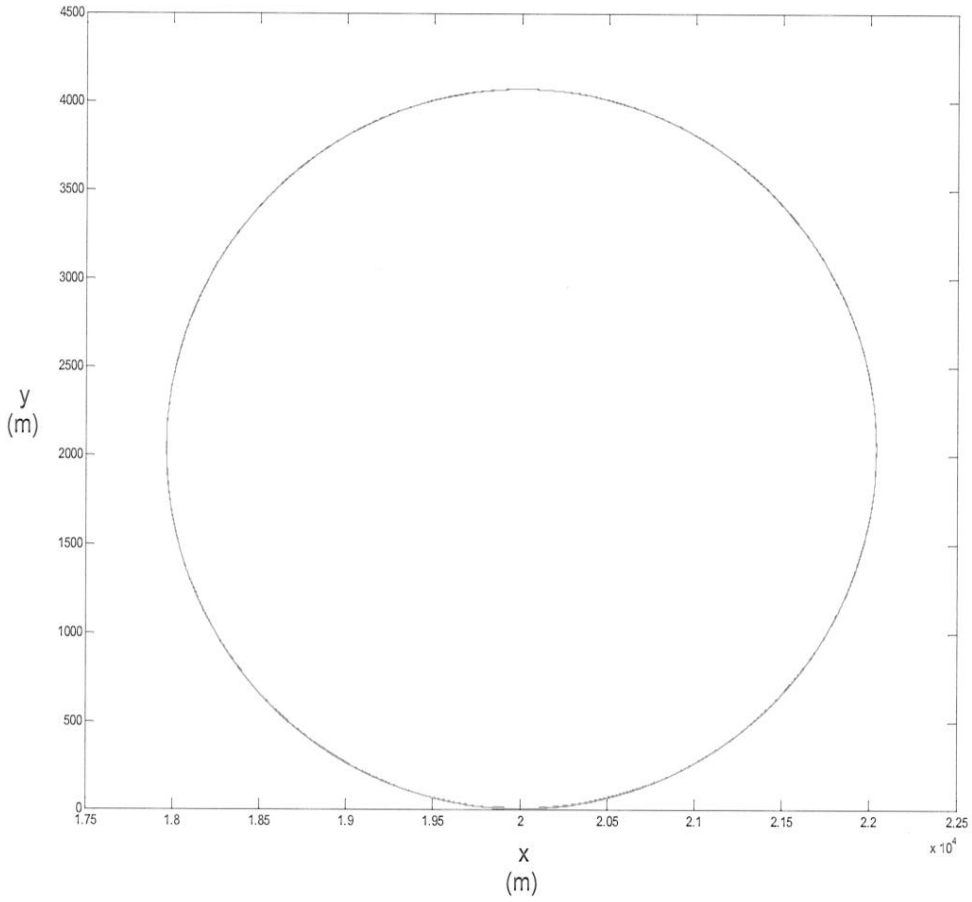


Figure 6: Aircraft Kinematic Model. The trajectory is a circle with radius (V/Ω). True (red) vs. Estimated (blue) trajectory is shown (one Monte Carlo run).

This figure displays the strength of the input estimation-based MWA. The tracking filter performs very well.

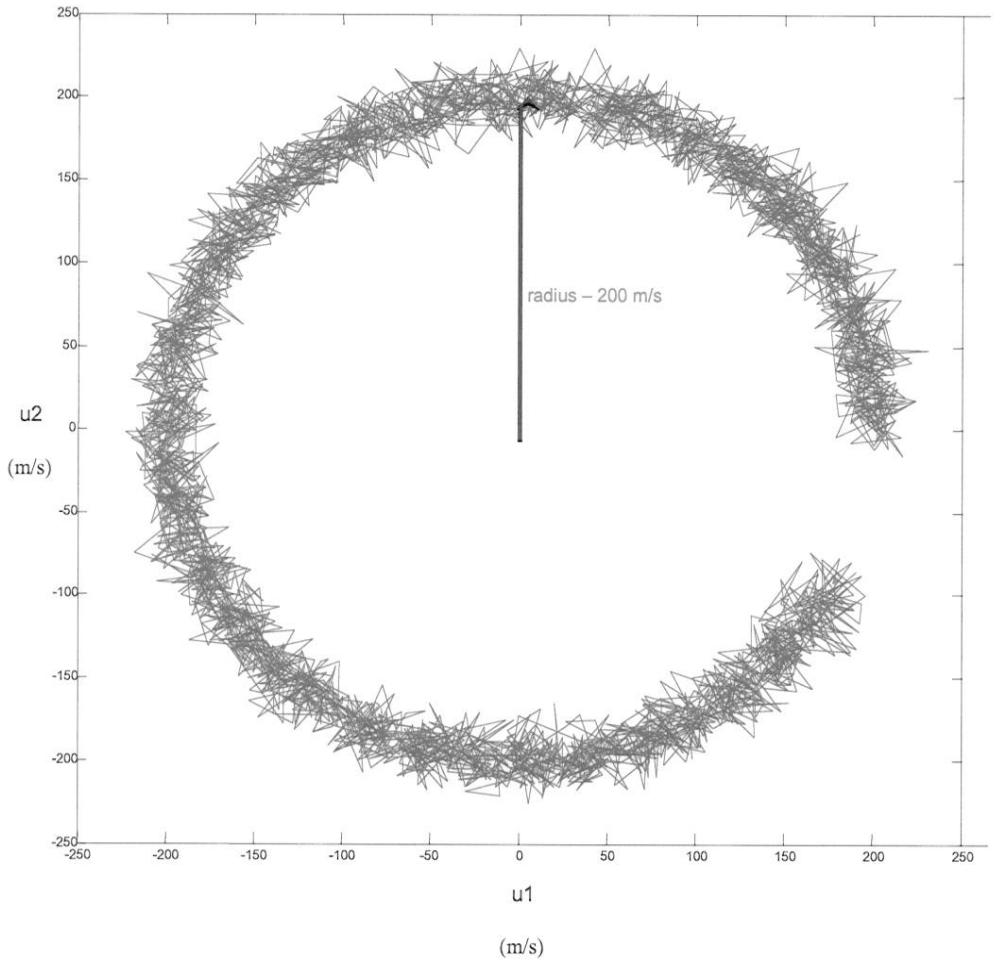


Figure 7: Aircraft Kinematic Model. Estimated Input Level. u_2 vs. u_1 plot (one Monte Carlo run).

This figure contains one Monte Carlo run and shows the result of high noise intensity corrupted measurements. Even with such high noise characteristics, the control signal can be efficiently determined from this graph. The aircraft's velocity, $\hat{V}$, is estimated by obtaining the radius of the circle shown ($\hat{V} \approx \sqrt{\hat{u}_1^2 + \hat{u}_2^2}$).

4.1.4.1 Maneuvering Aircraft Simulation

We now wish to see how well the algorithm performs against a maneuvering aircraft with variable velocity and turn rate. The Maneuvering Aircraft Simulation was tested using the parameters in Table 5. The parameters are the same as for the previous aircraft kinematic model with the exception of changing the aircraft maneuver every 1000 time steps.

Table 5: Parameters for Maneuvering Aircraft Simulation

$0 \le t < 1000$	$1000 \le t < 2000$	$2000 \le t < 3000$	$3000 \le t < 4000$
$V_{true} = 200$ m/s	$V_{true} = 100$ m/s	$V_{true} = -300$ m/s	$V_{true} = 50$ m/s
$\Omega = 0.098$ rad/sec	$\Omega = 0.2$ rad/sec	$\Omega = 0.3$ rad/sec	$\Omega = 0.5$ rad/sec

The results of the maneuvering aircraft simulation are presented in Figures 8-12.

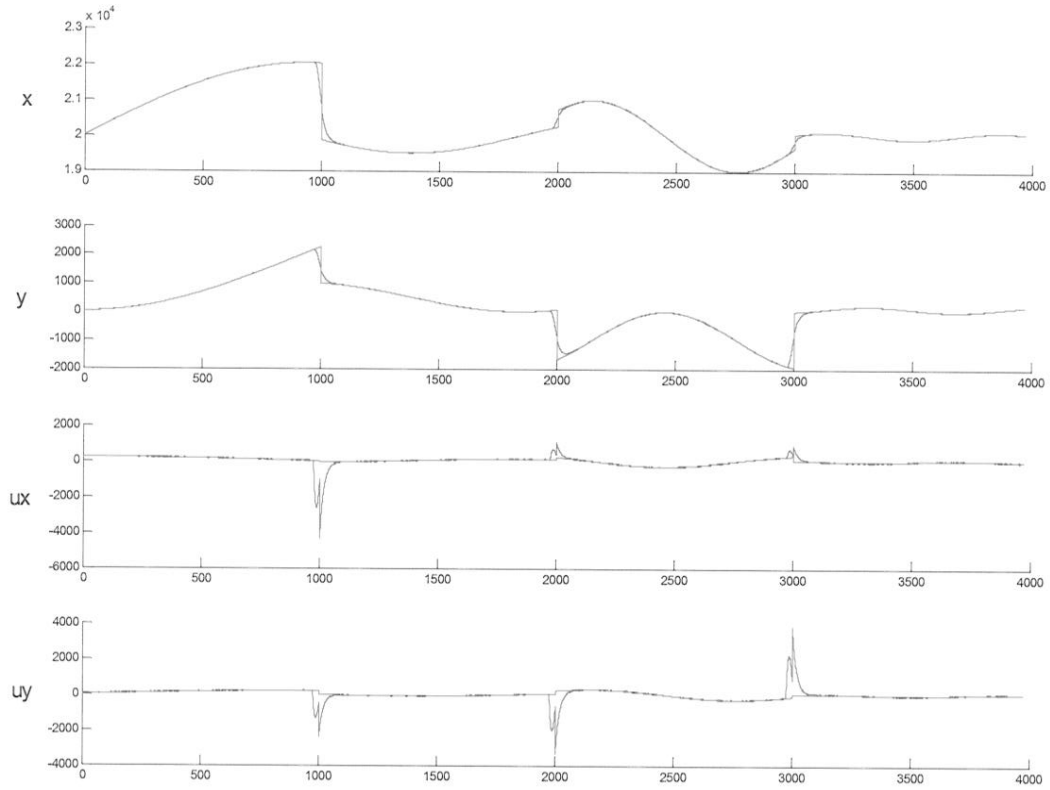


Figure 8: Maneuvering Aircraft States. True (red) vs. Estimated (blue). (one Monte Carlo run)

Figure 8 clearly displays an added benefit of this moving/sliding window algorithm: the spikes in the control signal's estimate indicate a discontinuity in the aircraft's control signal. Indeed, the filter assumes that the control signal is constant in the window. Since a jump in u occurs in the window, the estimates of u are bad and spike. A spike is a tell-tale sign of a maneuver onset. Hence, one can detect maneuvers using parameter estimation. Such detection ability allows you to declare points where the "constant u" model is not adequate. One should not try to use an estimate based on a window that

includes this point of discontinuity. Better yet, one should process up to that point in time, but not beyond, and then process separately from that point forward, seeking a different constant value for u through the estimator.

Figure 9 illustrates the aircraft being tracked despite a complex maneuver. Even though the aircraft undergoes unrealisticly sharp turns, the input-signal-estimation-based MWA tracking filter maintains track.

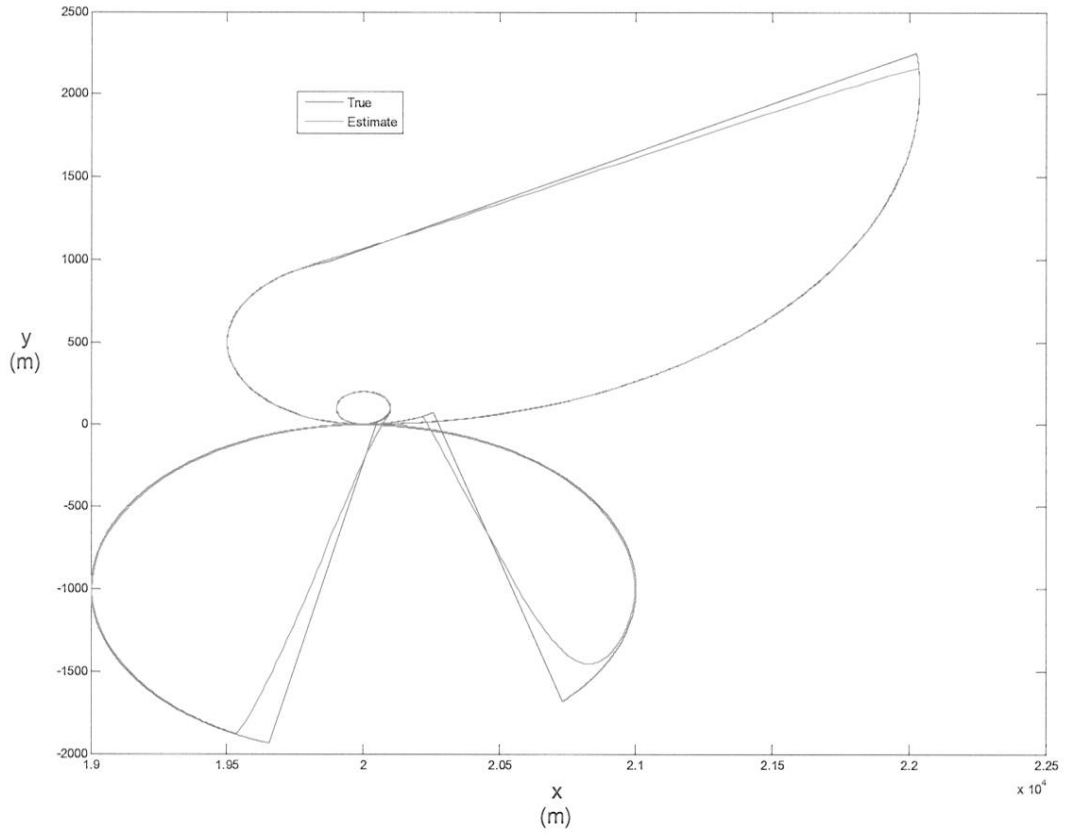


Figure 9: Maneuvering Aircraft Simulation. True vs. Estimated Complex Trajectory. (one Monte Carlo run)

Figures 10-12 further illustrate how well the MWA performed in the maneuvering aircraft simulation.

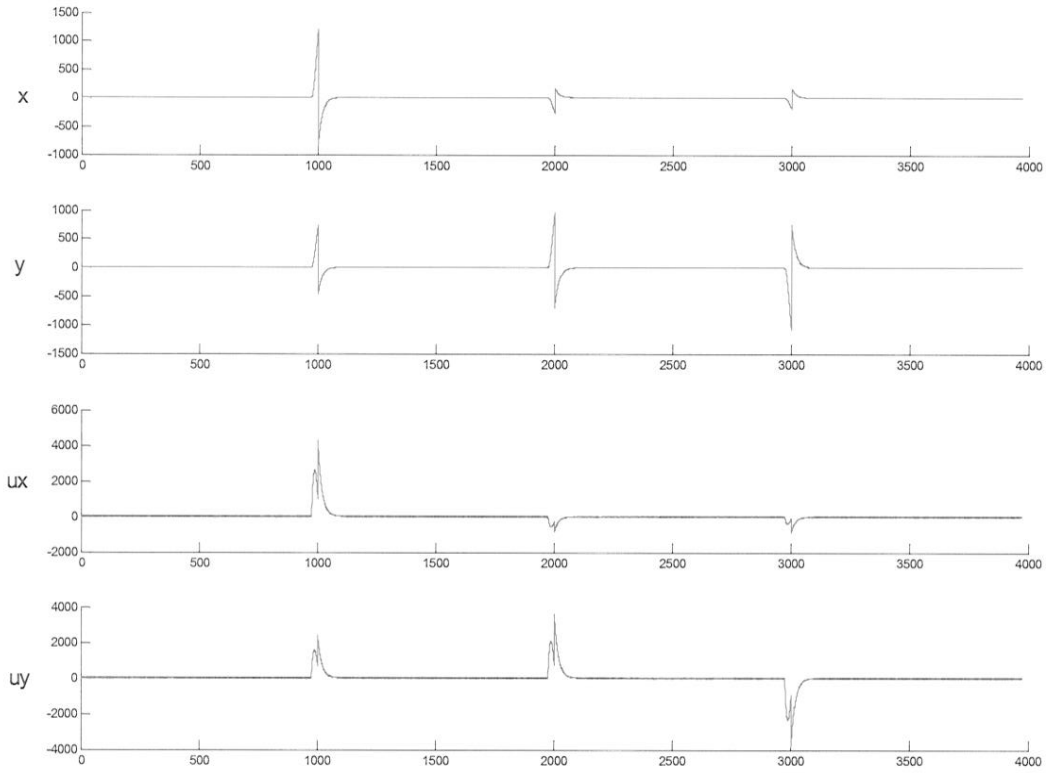


Figure 10: Mean Error +/- One Sigma (Monte Carlo evaluated). Error is more clearly seen in Figure 11. (10 Monte Carlo runs)

Once again, the figure shows spikes every 1000 time steps at the beginning of each maneuver. The tracking filter takes roughly 1-2 seconds to regain track after the onset of a new maneuver.

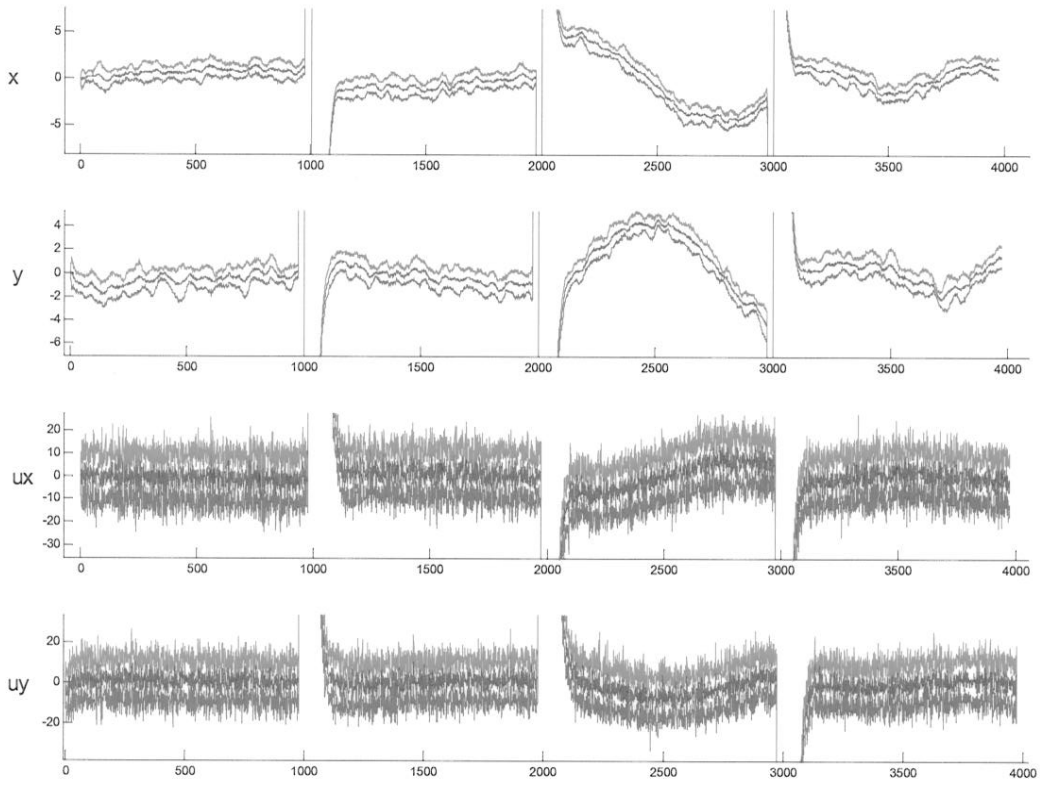


Figure 11: Zoomed in Mean Error +/- One Sigma (Monte Carlo evaluated). (10 Monte Carlo runs)

Position errors are small and the control signal's estimates are reasonable despite such high measurement noise injected into the simulation.

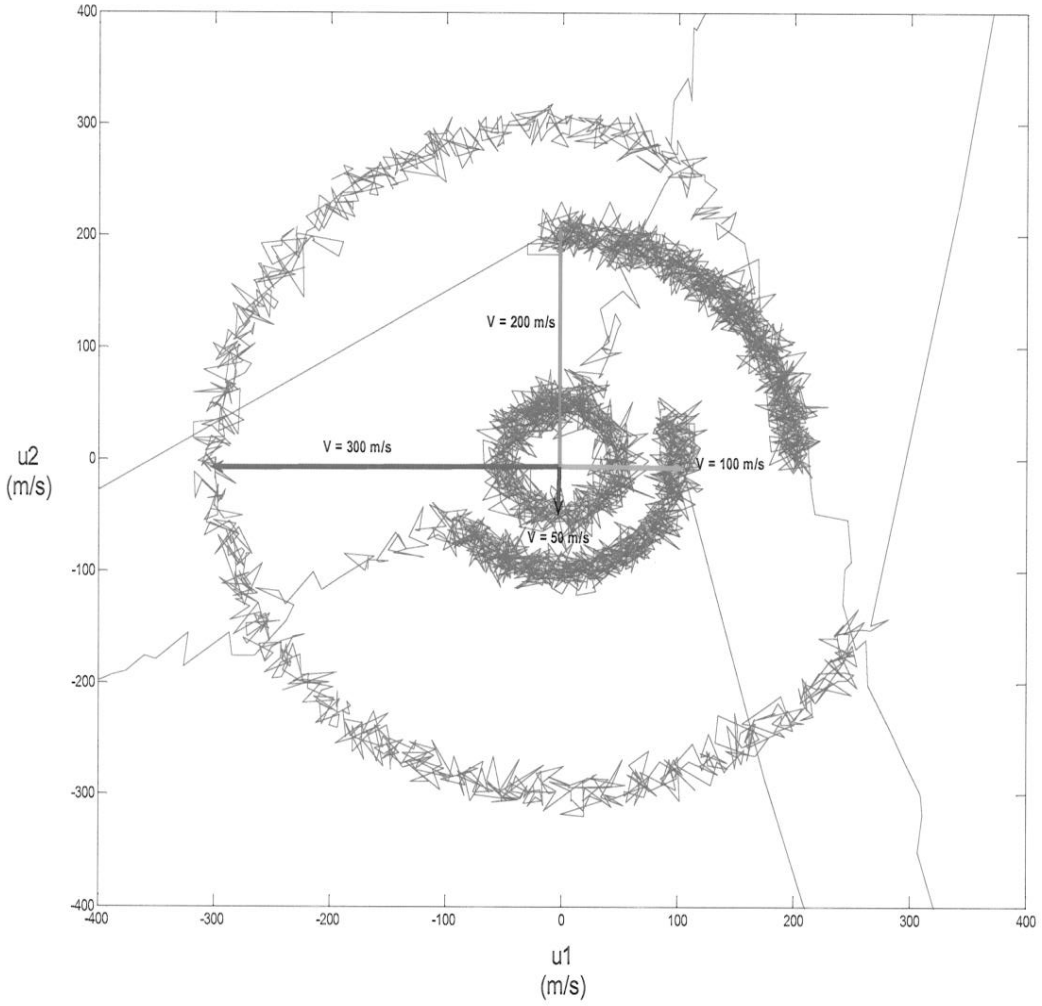


Figure 12: Estimated Control Signal. As before, the aircraft's velocity of each change in maneuver is indicated by the radius of each circle. (one Monte Carlo run)

Figure 12 is the estimated control signal from one Monte Carlo run of the maneuvering aircraft simulation. The erratic lines represent the MWA regaining track after each step maneuver change. Realistically, an aircraft's velocity will not jump from 200 m/s to 100 m/s in one *dt*, but even with such a step change in maneuver intensity, the input

estimation-based MWA handles it well. Moreover, it only takes about one second to determine the aircraft's current velocity of the subsequent maneuver.

4.1.5 Isotropic Rocket Model

The purpose of using the Isotropic Rocket model is to see how well the MWA estimates a target's acceleration state. We now consider the isotropic rocket kinematic model for a maneuvering target

$$\ddot{x} = u_x$$
$$\ddot{y} = u_y$$

let

$$v_x = \dot{x} \qquad\qquad v_y = \dot{y}$$

The augmented state vector

$$X \equiv \begin{pmatrix} x \\ v_x \\ u_x \\ y \\ v_y \\ u_y \end{pmatrix} \in \Re^6$$

The dynamics are

$$\dot{X} = AX$$
$$y = CX$$

$$X(0) = X_0 = \begin{pmatrix} x_0 \\ v_{x_0} \\ u_x \\ y_0 \\ v_{y_0} \\ u_y \end{pmatrix} \in \Re^6$$

where

$$A=\begin{bmatrix}0&1&0&0&0&0\\0&0&1&0&0&0\\0&0&0&0&0&0\\0&0&0&0&1&0\\0&0&0&0&0&1\\0&0&0&0&0&0\end{bmatrix}, \quad C=\begin{bmatrix}1&0&0&0&0&0\\0&0&0&1&0&0\end{bmatrix}$$

If Doppler measurements are also available, the measurement matrix is

$$C=\begin{bmatrix}1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&0&1&0&0\\0&0&0&0&1&0\end{bmatrix}$$

To check for observability, again calculate the observability matrix O and verify rank(O).

$$O=\begin{bmatrix}C^T & A^TC^T & \left(A^T\right)^2C^T\end{bmatrix}_{6\times 6}$$

Indeed in each case (Doppler or not), rank(O) = 6 and the system is observable. One can now solve the ensuing linear regression in the parameter

$$\vartheta=\left(x_0,v_{x_0},u_x,y_0,v_{y_0},u_y\right)^T$$

In fact, two decoupled linear regressions in $\left(x_0,v_{x_0},u_x\right)^T$ and $\left(y_0,v_{y_0},u_y\right)^T$ are obtained.

The Isotropic Rocket Model was put to the test using the parameters in Table 6.

Table 6: Parameters for Isotropic Rocket Model Simulation

$u_{x_{\max}}=41.54\ m/s^2$	$v_x=200\ m/s$	$x_0=0\,m$	$dT=0.3$ sec
$u_{y_{\max}}=41.54\ m/s^2$	$v_y=200\ m/s$	$y_0=0\,m$	$dt=0.01$ sec
$\sigma_k=0.1\,m$	$\beta=30$		

Recall that dT is the length (sec) of each estimation window and dt is the discretization time step (sec) of the moving Kalman filter. The results of the simulations are shown in Figures 13 and 14.

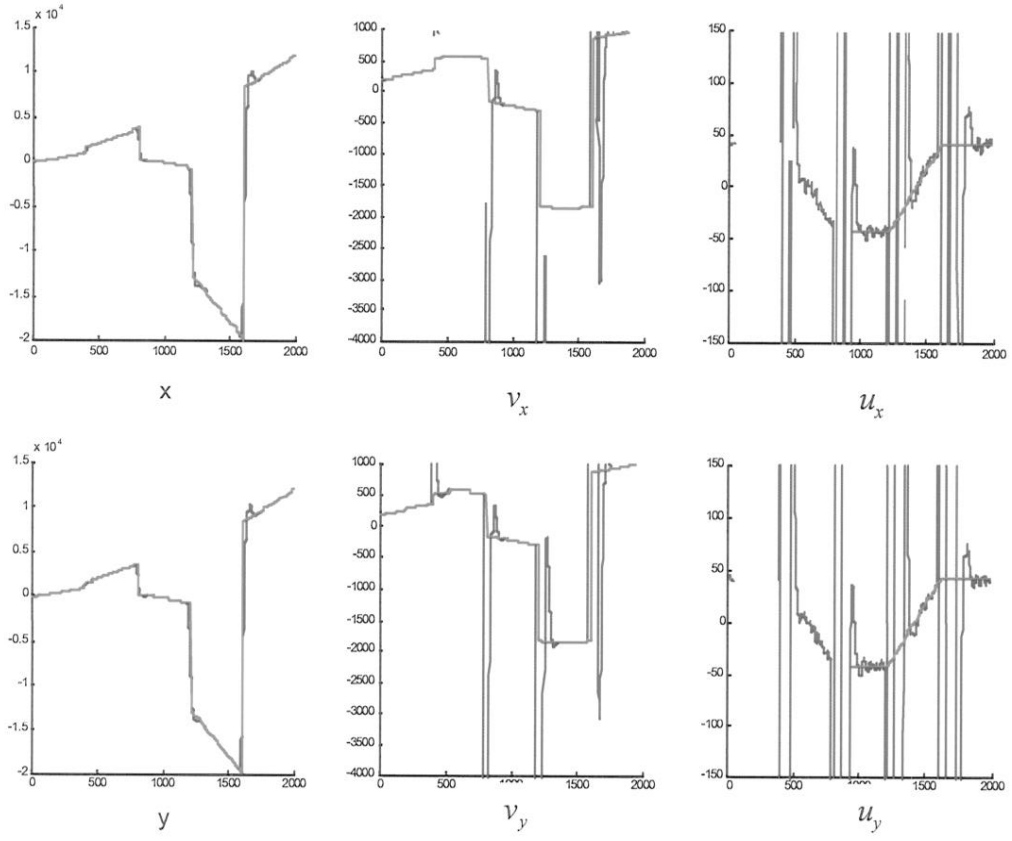


Figure 13: Isotropic Rocket Model States. True (red) vs. Estimated (blue) for $(x, v_x, u_x, y, v_y, u_y)$. (one Monte Carlo run)

Despite the step in the control signal, the algorithm maintains good track. The input estimate's spikes result from the step input changes alternating between negative and positive $41.5\, m/s^2$. This simulates a 4 g jinking maneuver performed by the aircraft.

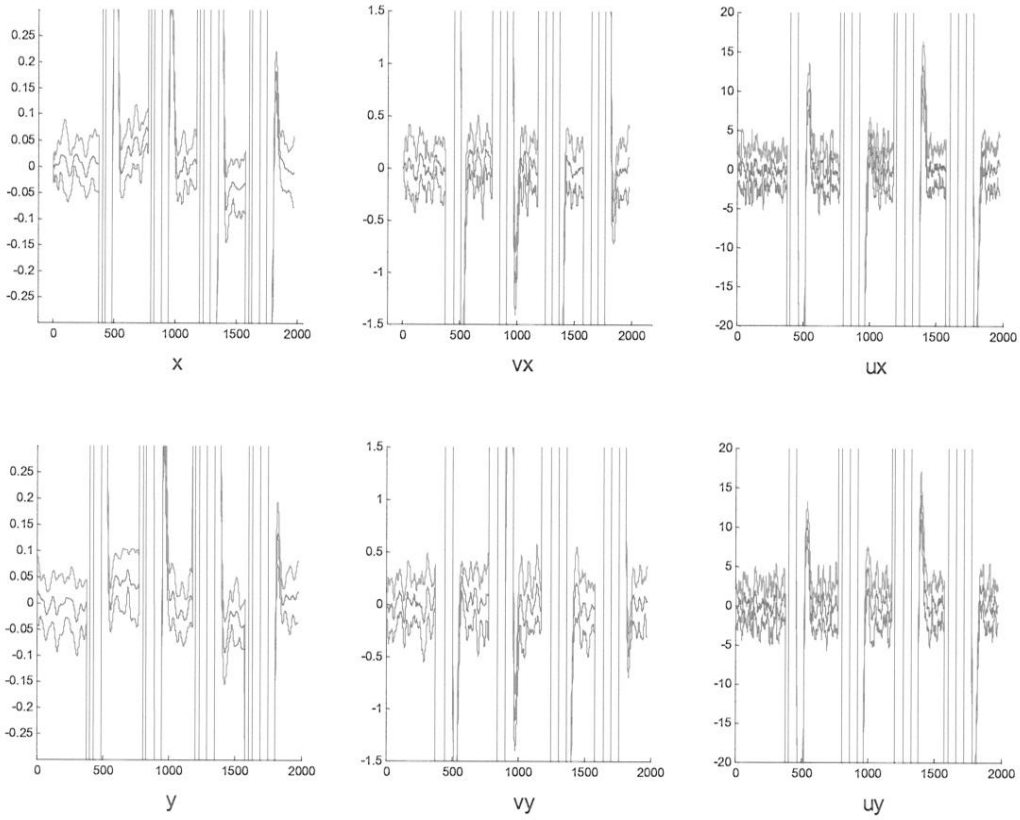


Figure 14: Isotropic Rocket Model. Mean Error +/- One Sigma (Monte Carlo evaluated) of Parameter Estimation error (10 Monte Carlo Runs). The track is recovered after 1.5 seconds.

The aircraft's position and velocity are tracked with great accuracy. This shows that periodic high amplitude step maneuvers are handled well by the input-estimation-based MWA.

4.1.5.1 Isotropic Rocket Model: Time-Varying Smooth Inputs

The purpose of this model is to extend the Isotropic Rocket model to sinusoidal time-varying smooth inputs. We consider the smooth Isotropic Rocket inputs

$$u_x(t) = A\sin(\omega t)$$
$$u_y(t) = A\cos(\omega t)$$

The truth model used is

$$x(t) = x_0 + v_{x_0}t - \frac{A}{\omega^2}\sin(\omega t)$$
$$v_x(t) = v_{x_0} - \frac{A}{\omega}\cos(\omega t)$$
$$y(t) = y_0 + v_{y_0}t - \frac{A}{\omega^2}\cos(\omega t)$$
$$v_y(t) = v_{y_0} + \frac{A}{\omega}\sin(\omega t)$$

$$X_0 = \begin{pmatrix} x_0 \\ v_{x_0} \\ u_{x_0} \\ y_0 \\ v_{y_0} \\ u_{y_0} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ A \end{pmatrix}$$

The truth model parameters for this maneuvering target ($dt = 0.01$) are listed in Table 7:

Table 7: Parameters for Smooth Isotropic Rocket Model Inputs

$0 \le t < 1000$	$1000 \le t < 2000$	$2000 \le t < 3000$	$3000 \le t < 4000$
$A = 41.54\sqrt{2}$	$A = 22\sqrt{2}$	$A = 50\sqrt{2}$	$A = 10\sqrt{2}$
$\omega = \frac{\pi}{4}$ rad/sec	$\omega = \frac{\pi}{2}$ rad/sec	$\omega = \frac{\pi}{12}$ rad/sec	$\omega = \frac{\pi}{6}$ rad/sec

The results for the maneuvering Isotropic Rocket Model are illustrated in the Figures 15-18.

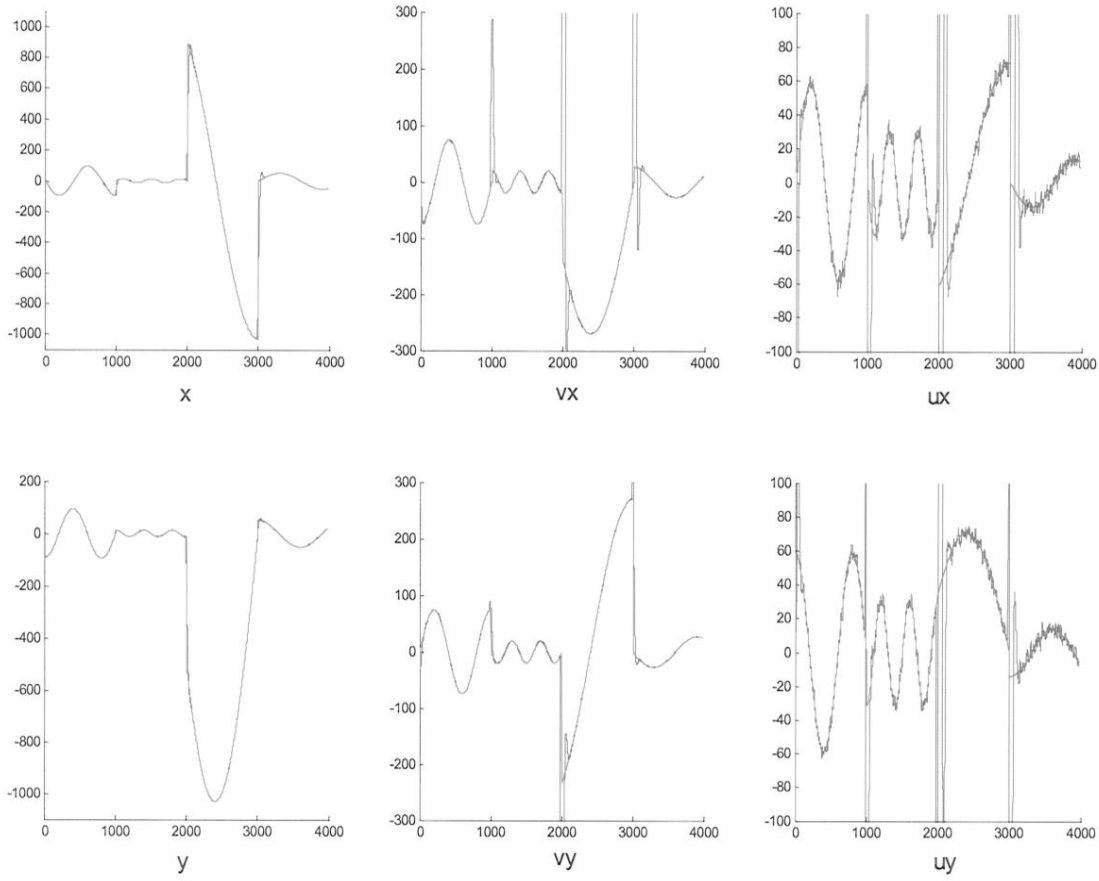


Figure 15: Smooth Isotropic Rocket Model True (red) vs. Estimated (blue) for $(x, v_x, u_x, y, v_y, u_y)$ (one Monte Carlo run)

Spikes in estimated trajectories (blue) indicate when a maneuver has begun. The MWA "detects" the maneuver no matter how strong it is and quickly regains track and produces good estimates of the target's velocity and input level, a.k.a., acceleration.

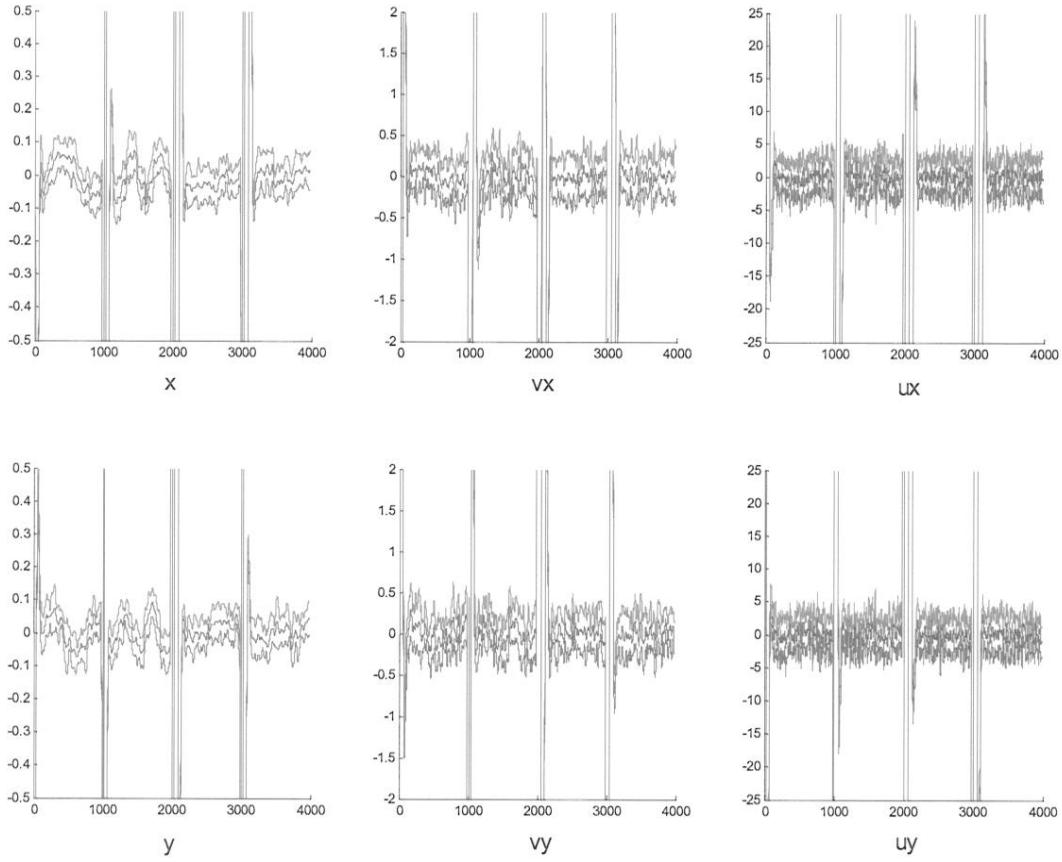


Figure 16: Smooth Isotropic Rocket Model Mean Error +/- One Sigma (Monte Carlo evaluated) (10 Monte Carlo runs)

It takes on the order of one-two seconds to regain track after each maneuver.

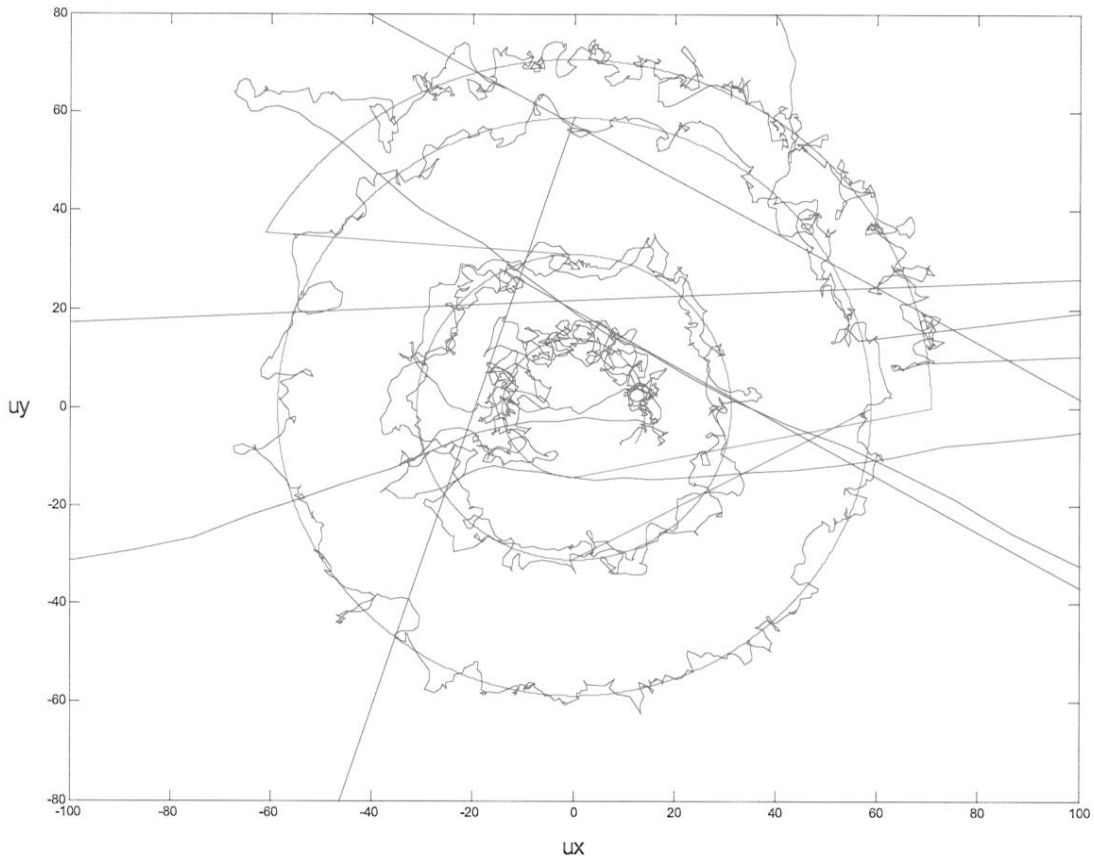


Figure 17: True Control Signal (red) vs. Estimated Control Signal (blue) (one Monte Carlo run)

Recall, the erratic lines in the figure indicate a step maneuver. The radii of each circle indicate the amplitude of the control input level for each maneuver. The estimated input level continues to track despite strong maneuvering of the aircraft.

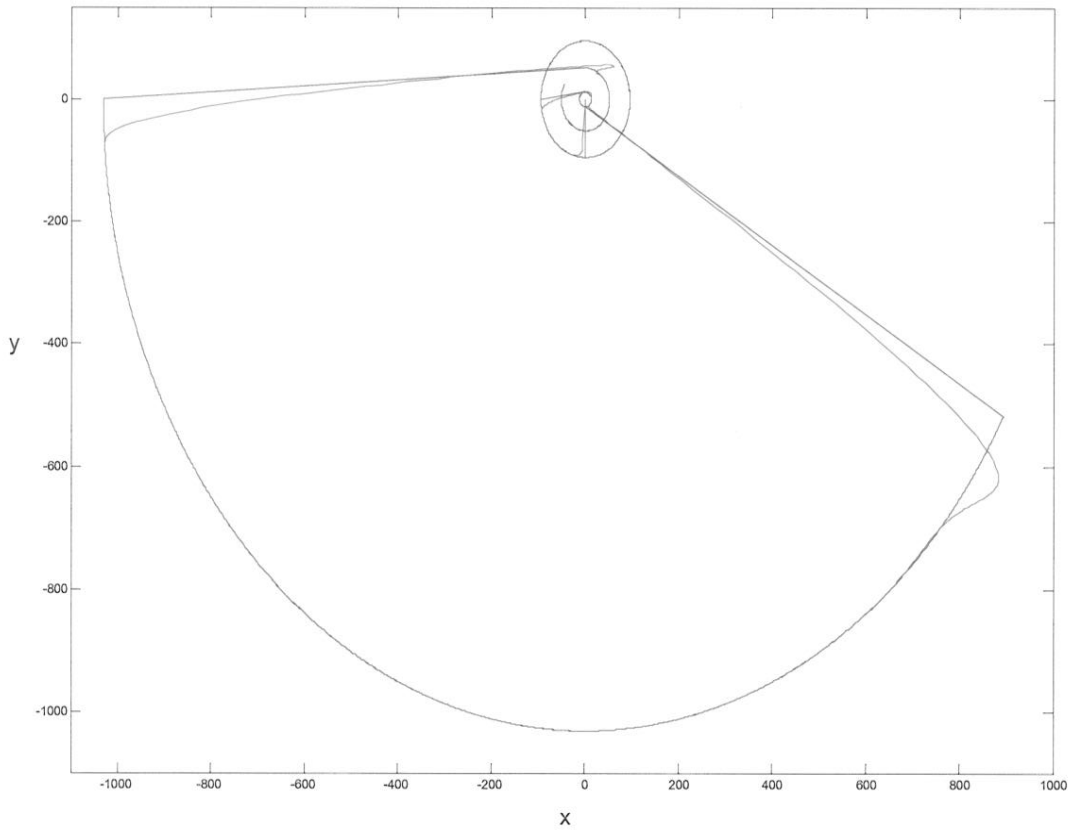


Figure 18: True (blue) vs. Estimated (red) Trajectory. Track is maintained. (one Monte Carlo run)

Figure 18 illustrates that the MWA can handle various turn rates. Another more challenging simulation was conducted using the truth model parameters in Table 8.

Table 8: Parameters for Complex Smooth Isotropic Rocket Model Inputs

$0 \le t < 1000$	$1000 \le t < 2000$	$2000 \le t < 3000$	$3000 \le t < 4000$
$A = 41.54\sqrt{2}$	$A = 22\sqrt{2}$	$A = 50\sqrt{2}$	$A = 10\sqrt{2}$
$\omega = \frac{\pi}{4}$ rad/sec	$\omega = \frac{\pi}{2}$ rad/sec	$\omega = \frac{\pi}{12}$ rad/sec	$\omega = \frac{\pi}{6}$ rad/sec
$v_{x_0} = 30\ m/s$	$v_{x_0} = 100\ m/s$	$v_{x_0} = 400\ m/s$	$v_{x_0} = -100\ m/s$
$v_{y_0} = 40\ m/s$	$v_{y_0} = 100\ m/s$	$v_{y_0} = 200\ m/s$	$v_{y_0} = 20\ m/s$

The results for the complex aircraft maneuvers are illustrated in Figures 19-22.

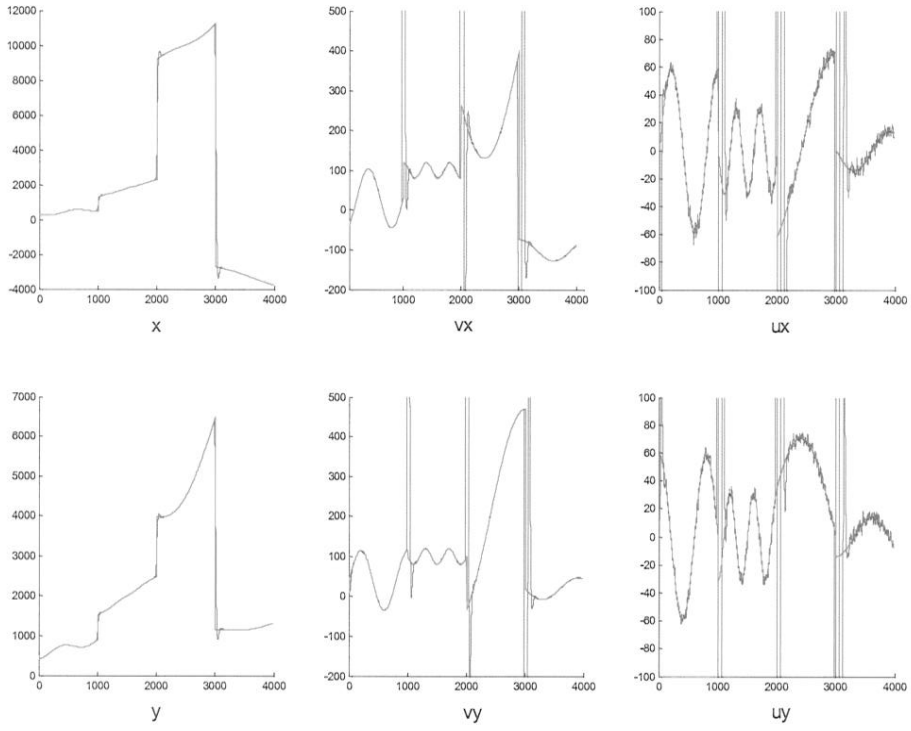


Figure 19: True (red) vs. Estimate (blue) for $(x, v_x, u_x, y, v_y, u_y)$ (one Monte Carlo run)

The algorithm still handles well such admittedly unrealistic step changes in maneuvera level. Despite varying the initial conditions, it still only requires a couple of seconds to regain track of the parameter. This is more clearly seen in Figure 20 where the mean error statistics are shown.

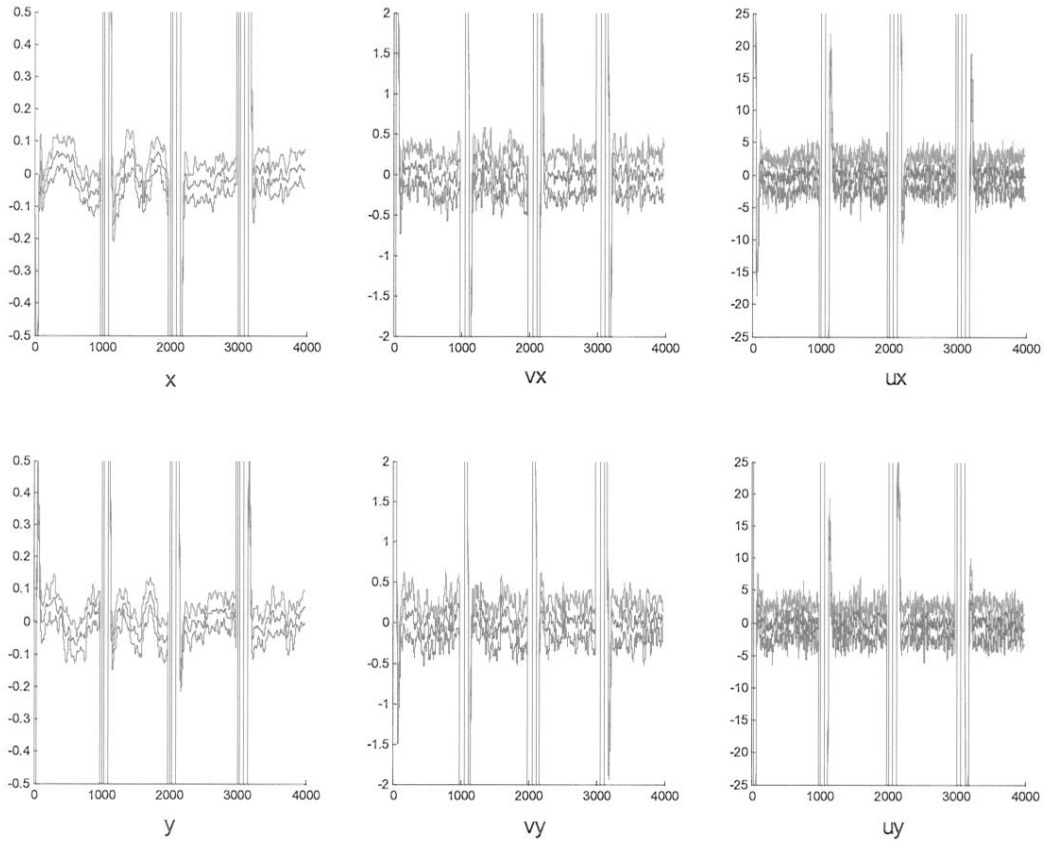


Figure 20: Mean Error +/- One Sigma (Monte Carlo evaluated) (10 Monte Carlo Runs). Again, RMS errors are favorable and it takes between 1-2 seconds to regain track.

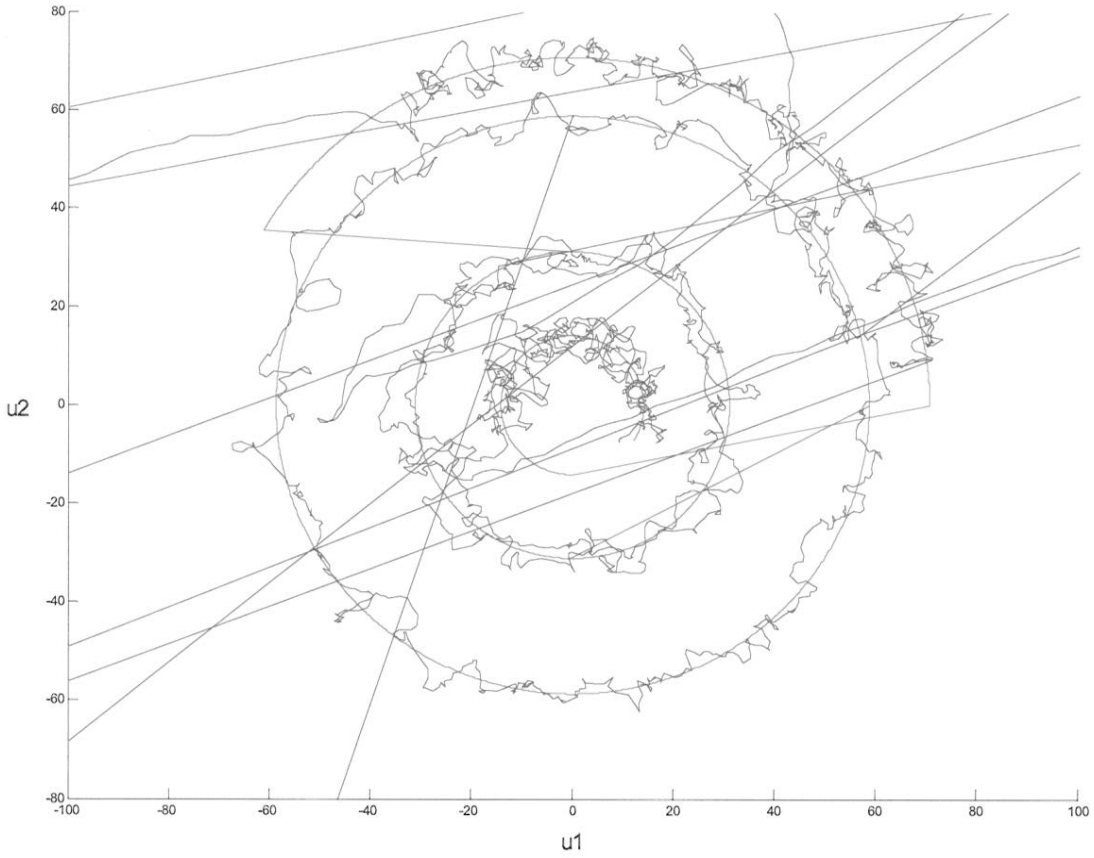


Figure 21: True Control Signal Level (red) vs. Estimated Control Signal Level (blue) (one Monte Carlo run).

The erratic lines in the Figure are caused by the MWA regaining track after each step maneuver change.

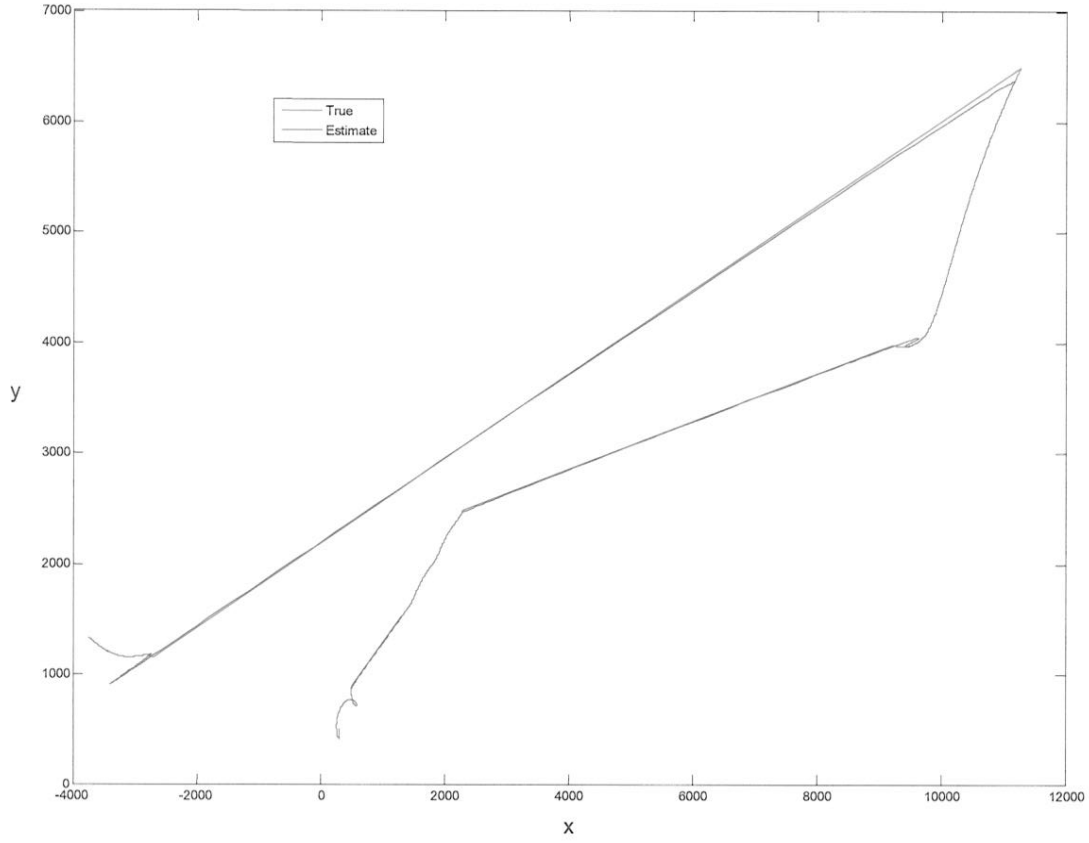


Figure 22: True vs. Estimated Aircraft Trajectory.

The aircraft remains tracked despite loops.

Now that the input-estimation-based MWA has performed considerably well against a variety of kinematic target models, the next objective is to measure its performance against a Multiple Model Adaptive Estimator (MMAE). First, the design of the MMAE is explained and then the seven-filter MMAE based estimator's performance is compared to simulation results obtained in the input estimation-based MWA.

4.2 MMAE Comparison

4.2.1 MMAE Fundamentals

The Multiple Model Adaptive Estimator uses a bank of Kalman filters, each based on a particular value of a parameter vector [4]. The parameter vector is a finite number of values ($a_1, a_2, \ldots, a_K$) as a result of discretizing a continuous parameter space. Associated with each parameter value is a different system model. The multiple model filtering algorithm and its structure is depicted in Figure 23.

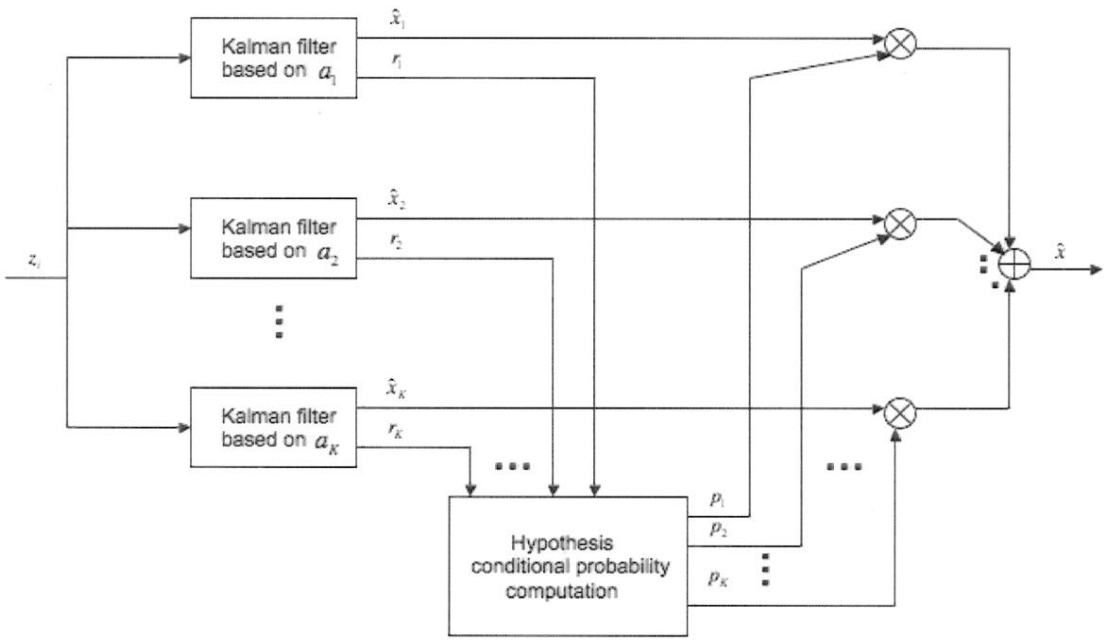


Figure 23. Multiple Model Filtering Algorithm [4]

When a measurement becomes available, the MMAE generates residuals for each Kalman filter, which are then passed on for further processing by a hypothesis

conditional probability computation. The conditional probabilities of each filter are used as weighting coefficients to generate the state estimate using the recursive relationship

$$p_k(t_i) = \frac{f_{z(t_i)|a,Z(t_{i-1})}(z_i \mid a_k, Z_{i-1}) p_k(t_{i-1})}{\sum_{j=1}^{K} f_{z(t_i)|a,Z(t_{i-1})}(z_i \mid a_j, Z_{i-1}) p_j(t_{i-1})}$$

Given the measurement history, this produces an overall state estimate,

$$\hat{x}(t_i^+) = \sum_{k=1}^{K} \hat{x}_k(t_i^+) p_k(t_i)$$

which is a probabilistically weighted average of the state estimates from each Kalman Filter in the bank using $p_k(t_i)$ as the appropriate weighting factors. The conditional covariance of $x(t_i)$ is

$$P(t_i^+) = \sum_{k=1}^{K} p_k(t_i) \left\{ P_k(t_i^+) + \left[\hat{x}_k(t_i^+) - \hat{x}(t_i^+)\right]\left[\hat{x}_k(t_i^+) - \hat{x}(t_i^+)\right]^T \right\}$$

Moreover, the estimate of the parameter vector (conditional mean of a at time t_i) is

$$\hat{a}(t_i) = \sum_{k=1}^{K} a_k p_k(t_i)$$

with a conditional covariance,

$$E\left\{\left[a - \hat{a}(t_i)\right]\left[a - \hat{a}(t_i)\right]^T \mid Z(t_i) = Z_i\right\} = \sum_{k=1}^{K} \left[a_k - \hat{a}(t_i)\right]\left[a_k - \hat{a}(t_i)\right]^T p_k(t_i)$$

The performance of this algorithm depends on the discretization of the parameter space and the MMAE's ability to distinguish between "correct" and "mismatched models". In other words, it must be able to distinguish between the residual characteristics of each Kalman filter.

For the sake of comparison with the MWA, an MMAE that minimizes the RMS position and velocity error against a benign and/or step-maneuvering target which cannot exceed 20-g maneuvers is used. The author in the class, EENG 768 at AFIT, Summer 2005, developed the MMAE based tracking filter.

4.2.2 MMAE Design

It must first be mentioned that this MMAE was not designed for this MWA comparison specifically. The MMAE chosen and the particular tuning used for each elemental filter were developed for a generic tracking filter and not with our current MWA simulations in mind. This comparison may not represent a fair comparison to "the best an MMAE could do with the classes of problems represented in this thesis". Nonetheless, the author cautiously attempted to compare the following MMAE to the MWA developed in this thesis.

A frequency domain (FD) model design procedure was systematically used to build a seven-model MMAE that could handle various benign and step-maneuvers. Three First Order Gauss-Markov Acceleration (FOGMA) filters were created to accommodate 1-, 6-, and 20-g benign maneuvers with correlation time coefficients of 4 seconds, 2 seconds, and 1 second, respectively. Step-maneuvers off these benign trajectories are tracked using three *Periodically Correlated Acceleration (PCA) Complement Filters*, which are designed to model the higher frequency content of the dynamics due to a step change in the target trajectory. These filters are termed "complement filters" because they work in conjunction with the FOGMA models as depicted in Figure 24. Each of the PCA Complement Filters has a peak which closely aligns with the cut-off frequency of each FOGMA filter so that step changes in target dynamics are modeled by the MMAE. The

seventh model is a Second Order Gauss-Markov Acceleration (SOGMA) filter, which is intended to handle very fast on-off step maneuvers. It was chosen to capture those maneuvers performed in the higher frequency ranges and hopefully catch any other maneuver performed outside the scope of the other six filters used in the MMAE.

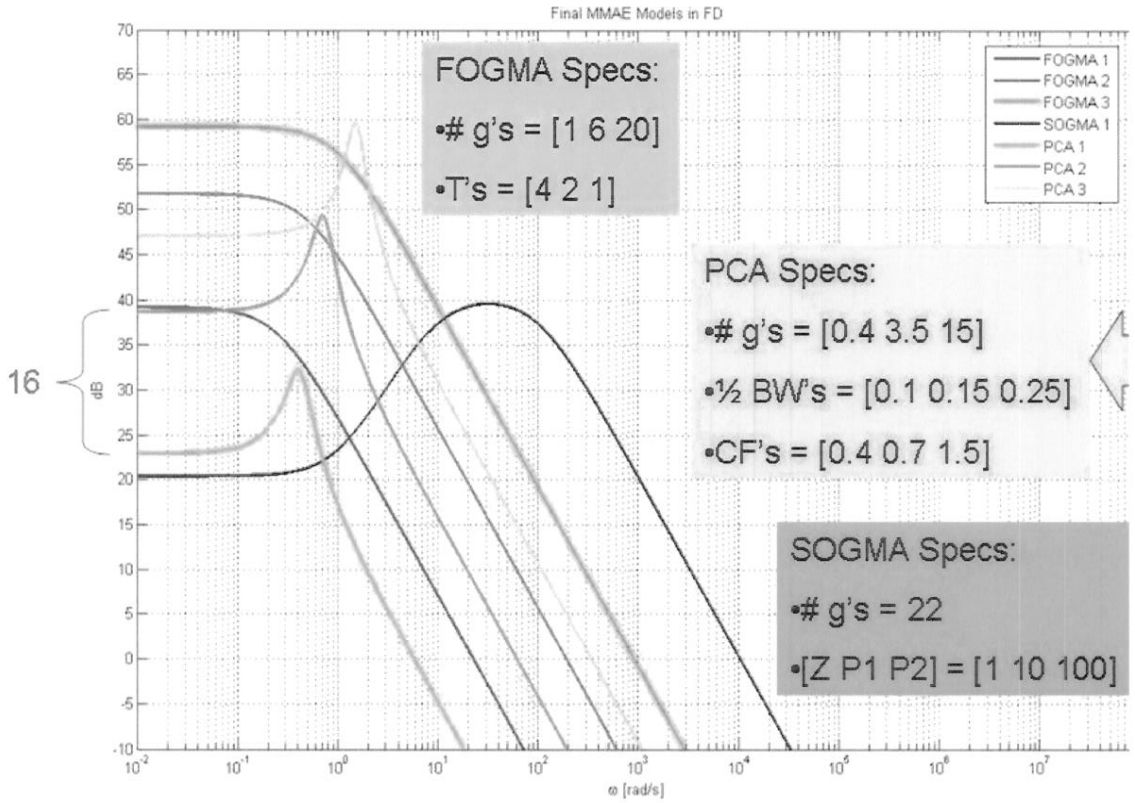


Figure 24: PSD plot of the final design of the MMAE models.

Finally, the design included the ad-hoc modifications of *Modal Probability Lower Bounding* (MPLB) and *Filter Re-starts* (FR) to allow the MMAE to track time-varying target dynamics. Altering the probability lower bounds prevented elemental filters from reaching zero allowing better probability flow between the elemental filters. The MPLB was set to the heuristic value provided in class of $0.1\times\{\#\text{ of filters}\}^{-1}$. Altering the

threshold on "restarting" diverged filters allowed the diverging filters to recover more rapidly. The FR threshold was set to 10 (or five times the approximate value of $r^T A^{-1} r$ (the residual, residual covariance inverse quadratic product). Divergent filters (filters which exceeded the FR threshold) were re-initialized with the MMAE state mean and covariance estimates (less the divergent filters). These two tuning parameters (MPLB and FR) are important in making the MMAE algorithm perform efficiently. Moving the probability to the filter with the "most correct" model is essential for a good MMAE making the "$r^T A^{-1} r$" term important as well. This value equates to the number of measurements (two for this design) for that filter in the MMAE, which matches the true trajectory. In other words, we want the probability to flow to the elemental filter in the MMAE closest to this number.

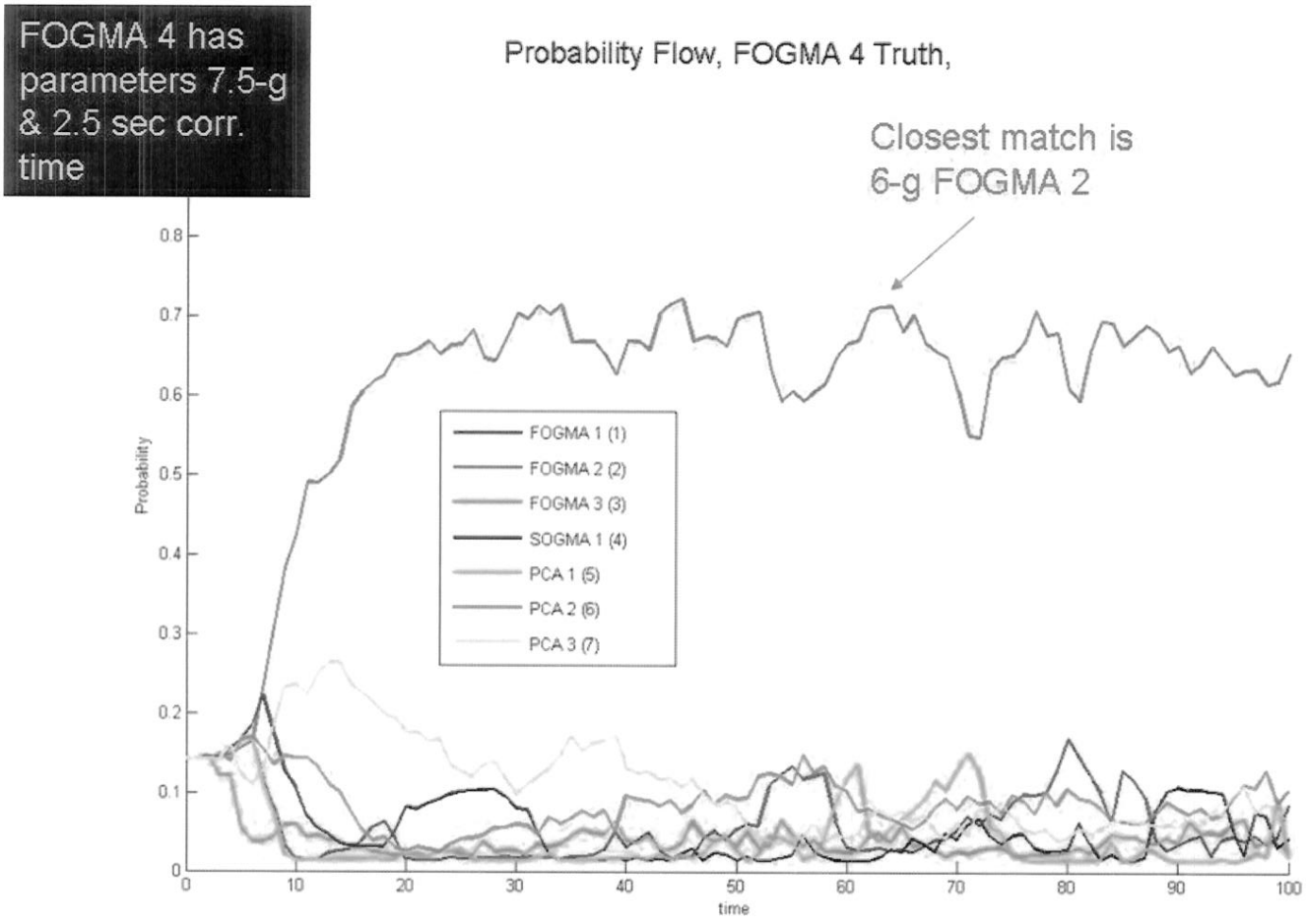
4.2.3 Design Test Results

The final MMAE design was tested against four scenarios. The following test cases are used to test the efficiency of the MMAE algorithm. They are only used to design a "good" MMAE for this generic tracking problem. Test Case 1 evaluated the performance of the MMAE against a truth model set to the parameters of one of the three FOGMA models without a step-maneuver. Test Case 2 allowed the truth model to be any FOGMA model not set to the parameters the MMAE's FOGMA models (again, without a step maneuver). Test Cases 3 and 4 are variations of Test Cases 1 and 2, respectively, in that a step maneuver occurs at some point in the simulation run. The first two test cases set the FR threshold to 200 while the last two test cases set it to 10 to make the MMAE. The FR threshold was set lower in Test Cases 3 and 4 to accommodate for the step maneuvers

performed in the simulation. The first test case used the FOGMA 1 model and provided good results as shown in Figure 25.

Figure 25: Modal probability flow averaged over 10 Monte Carlo simulations for each of the seven MMAE filters under Test Case 1 conditions with the truth model parameters set to FOGMA 1 values.

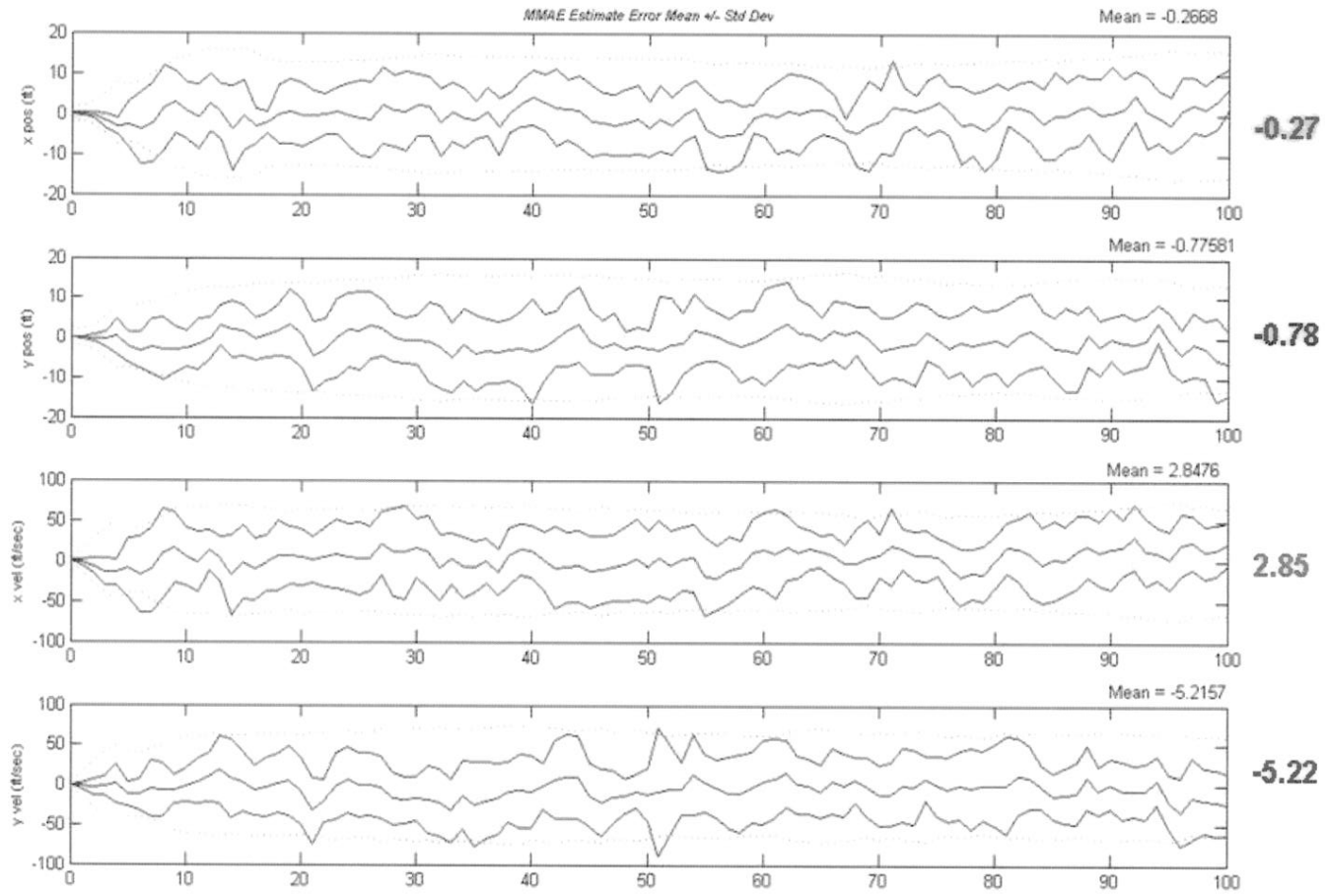
The proper filter (the one matched to truth) assumed most of the probability. Test Case 2 also produced expected results since the closest filter obtained the majority of the modal probability. The truth parameters and results are shown in Figure 26.



(a)

Figure 26: (a) Modal probability flow averaged over 10 Monte Carlo simulations

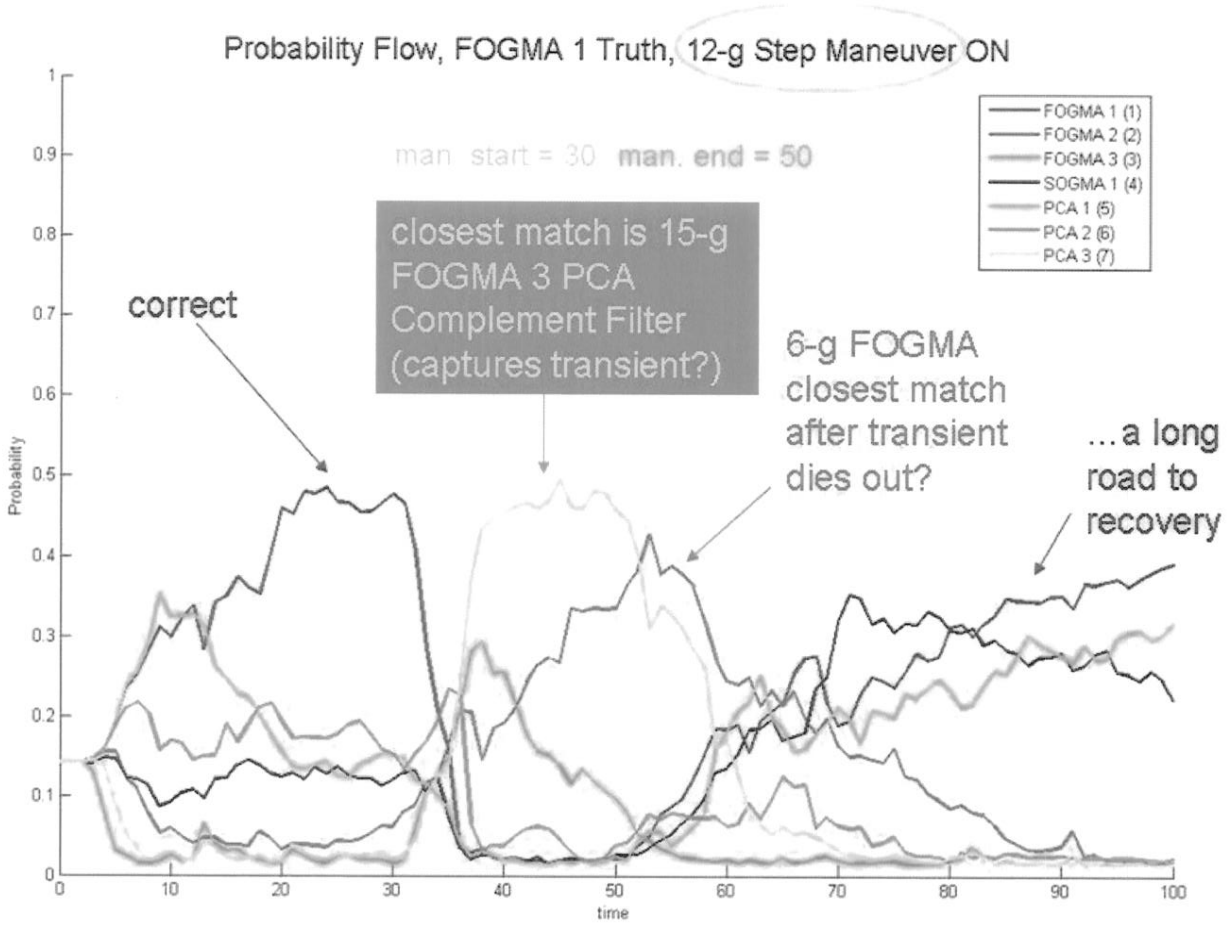
In the overall MMAE RMS position and velocity error plots, there are five traces. The middle three are the MMAE estimate error mean +/- one sigma (standard deviation) as computed from the Monte Carlo simulations. The top and bottom traces represent how well the MMAE "believes" it is doing in estimating the states. As long as the Monte Carlo statistics stay within the MMAE computed lines, this MMAE is performing as it should.



(b)

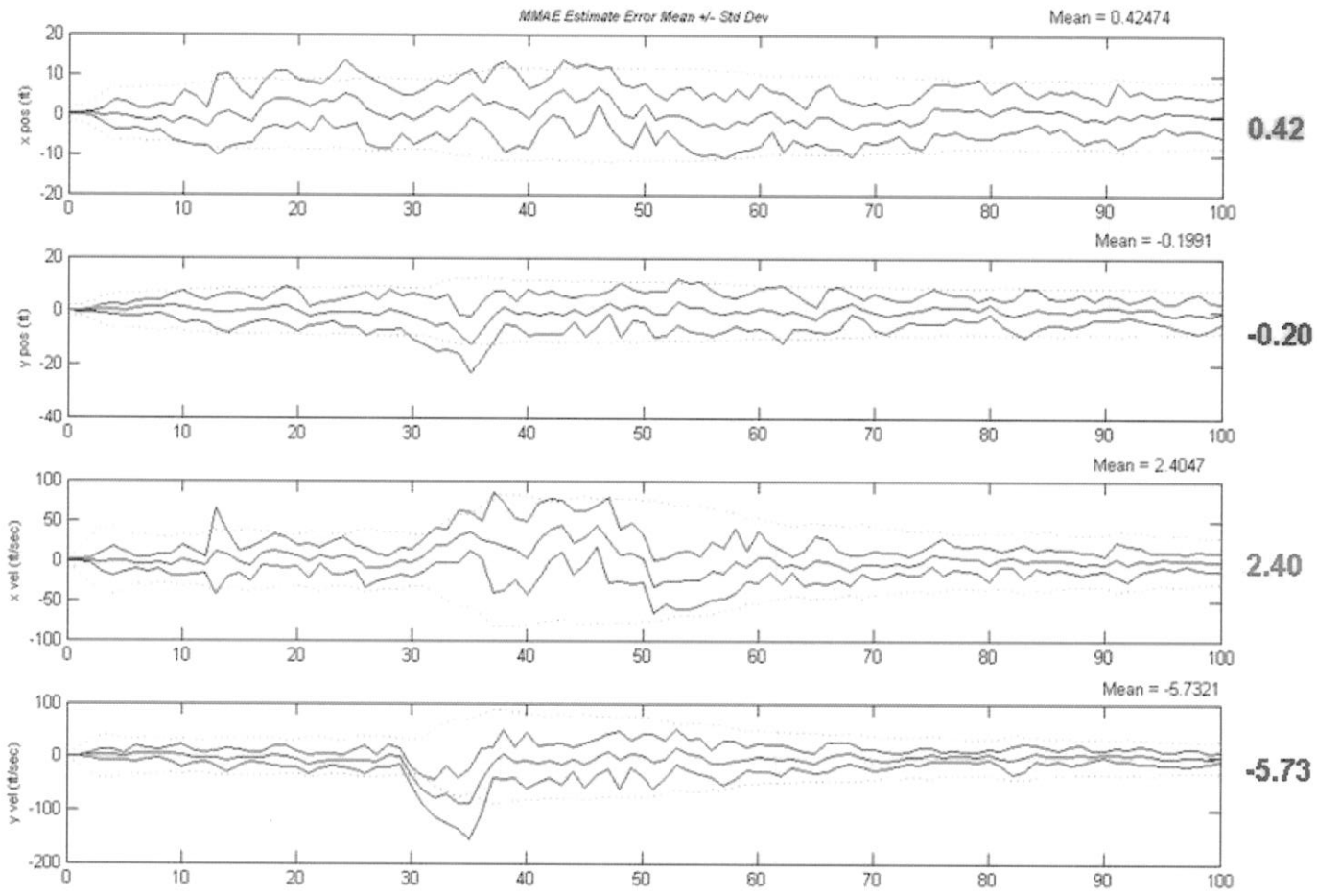
Figure 26: (b) Overall MMAE RMS position and velocity error

The third test case allowed a step maneuver, so the FR threshold was lowered to 10. The results of two sub-cases are depicted in Figure 27. Note that the PCA Complement Filters appear to model the initial (the step up or onset of maneuver) step change in target dynamics and not the final step change (the step down or termination of maneuver).



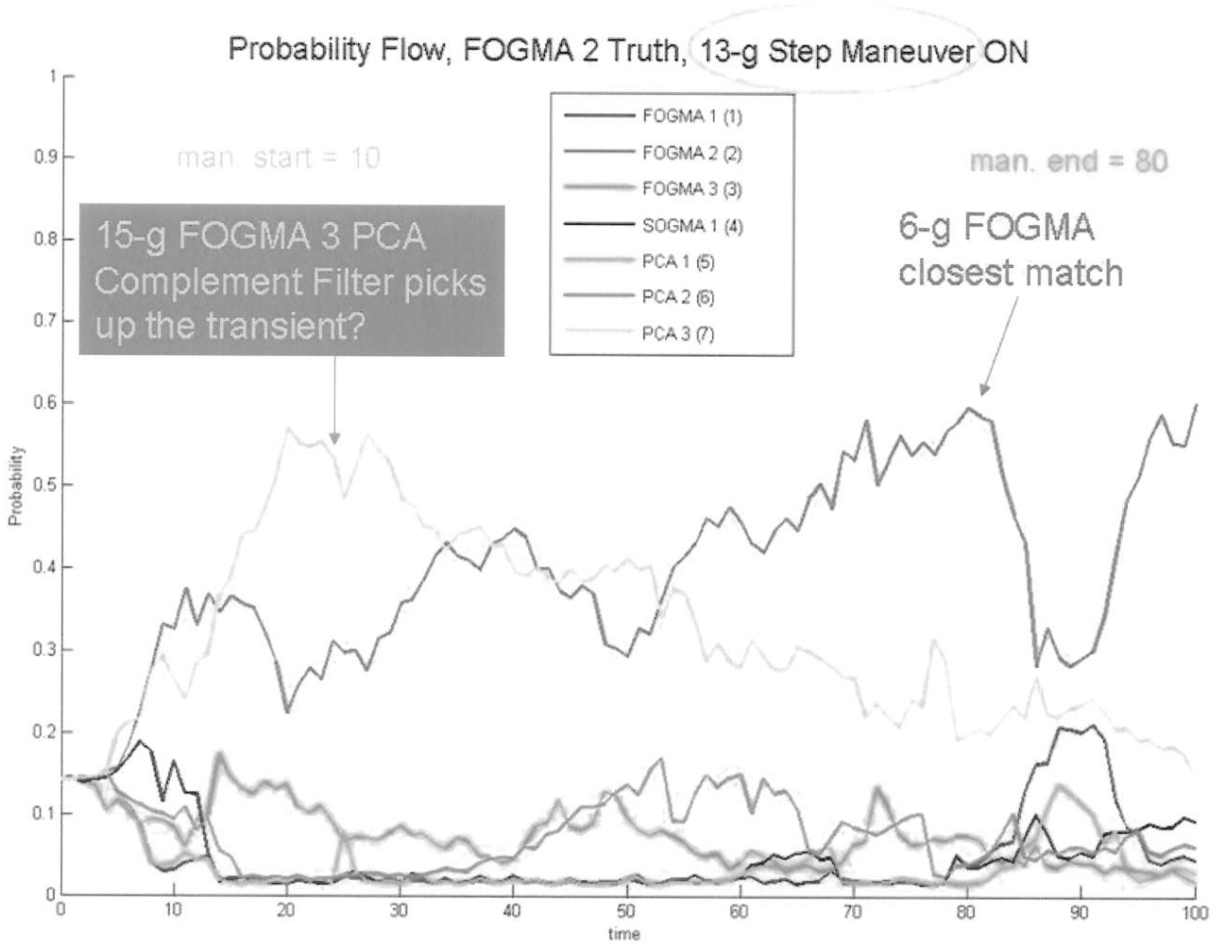
(a)

Figure 27: (a) Modal probability flow for each sub-case over 10 Monte Carlo Simulations



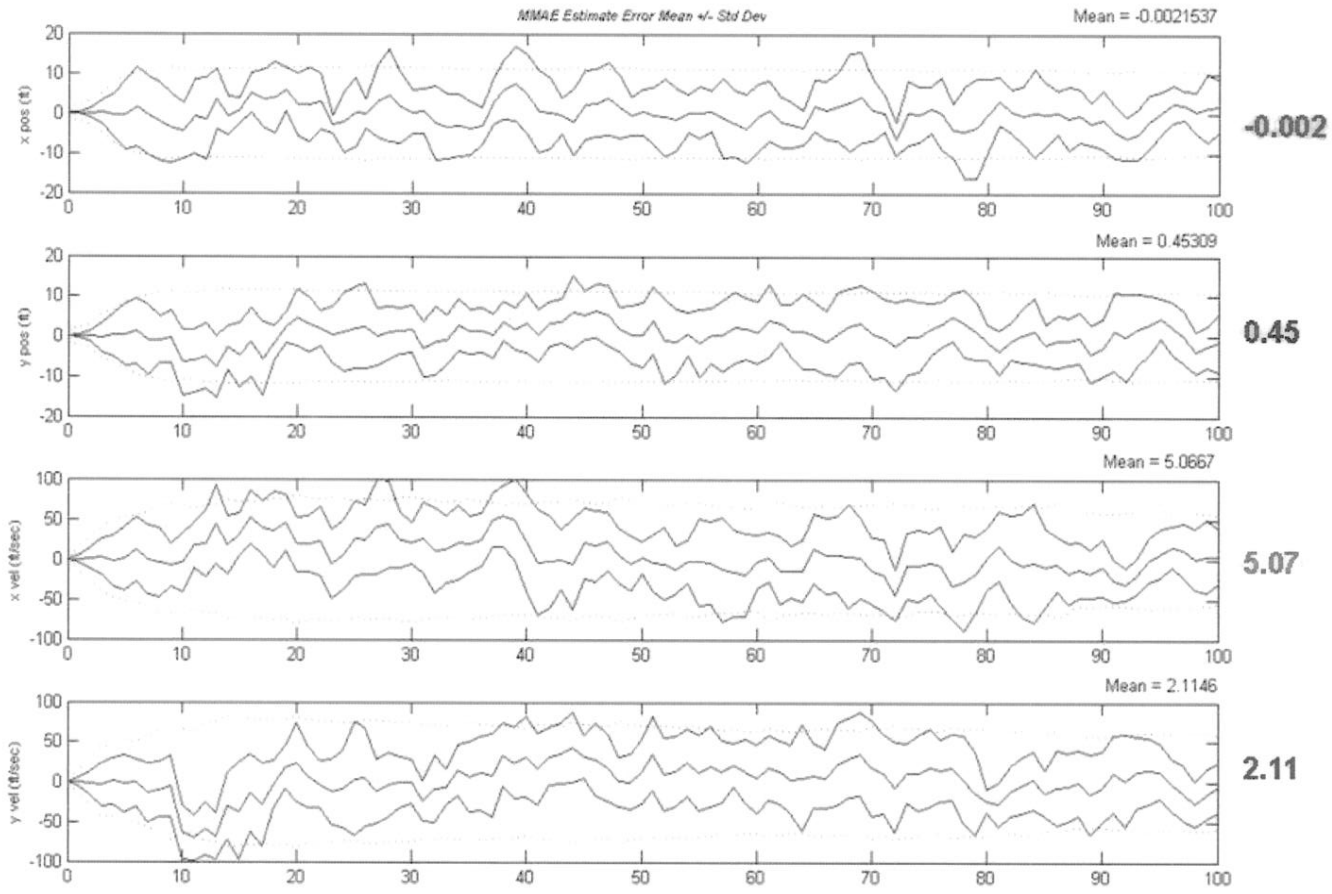
(b)

Figure 27: (b) Overall MMAE RMS position and velocity error



(c)

Figure 27: (c) Modal probability flow for each sub-case over 10 Monte Carlo Simulations

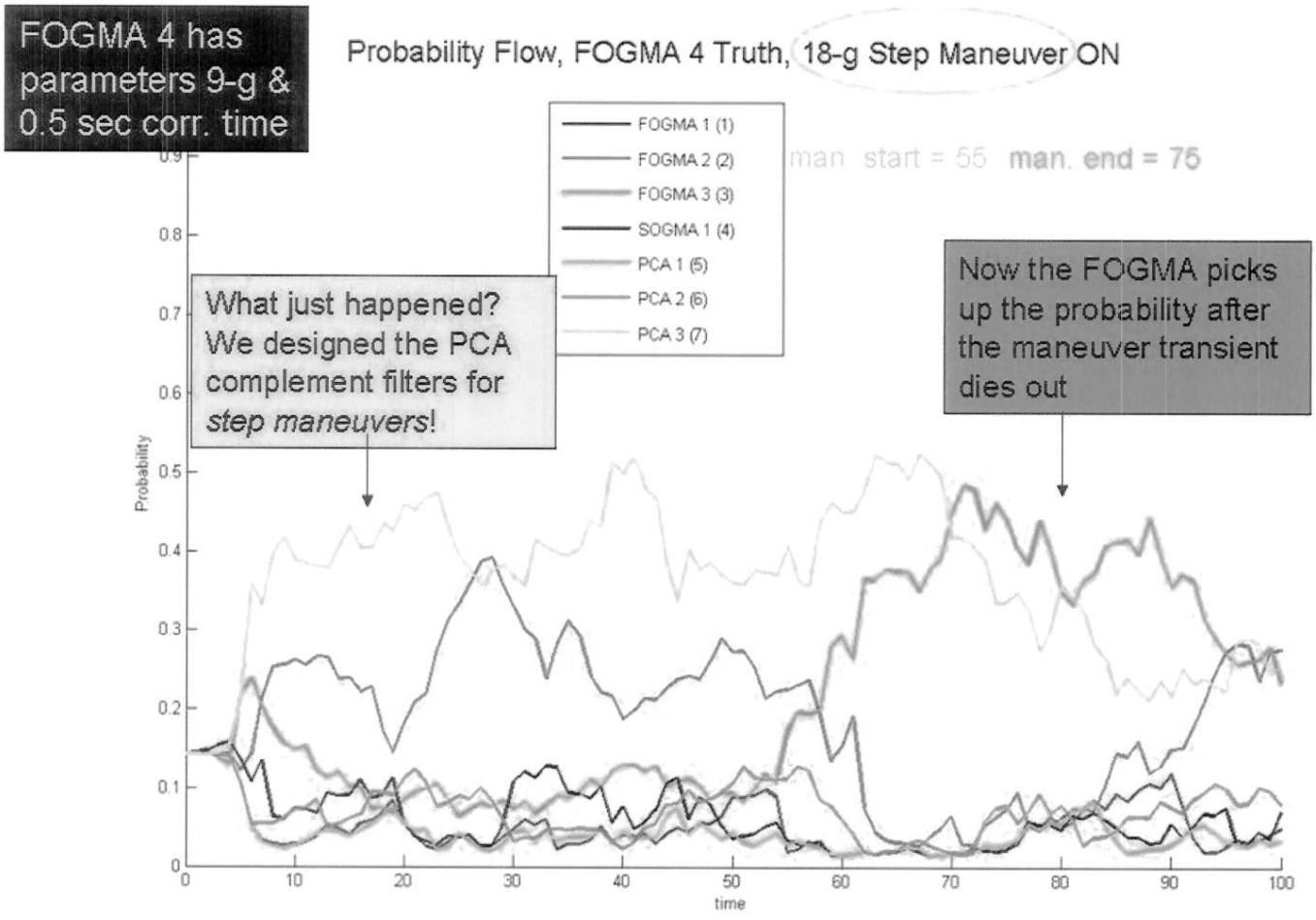


(d)

Figure 27: (d) Overall MMAE RMS position and velocity error

The results for Test Case 4 are shown in Figure 28. This case demonstrated that although the PCA Complement Filters were designed to handle step-maneuvers, these filters can also model benign maneuvers if their steady-state characteristics match that of the truth model. Referring to Figure 24, the steady-state mean square acceleration of all of the filters is given by the area under each filter's PSD curve (keep in mind that the PSD plot in Figure 24 does not show the negative frequency portion of the PSD curves,

so the area under each PSD curve shown in the figure is actually one-half of the steady-state RMS error).

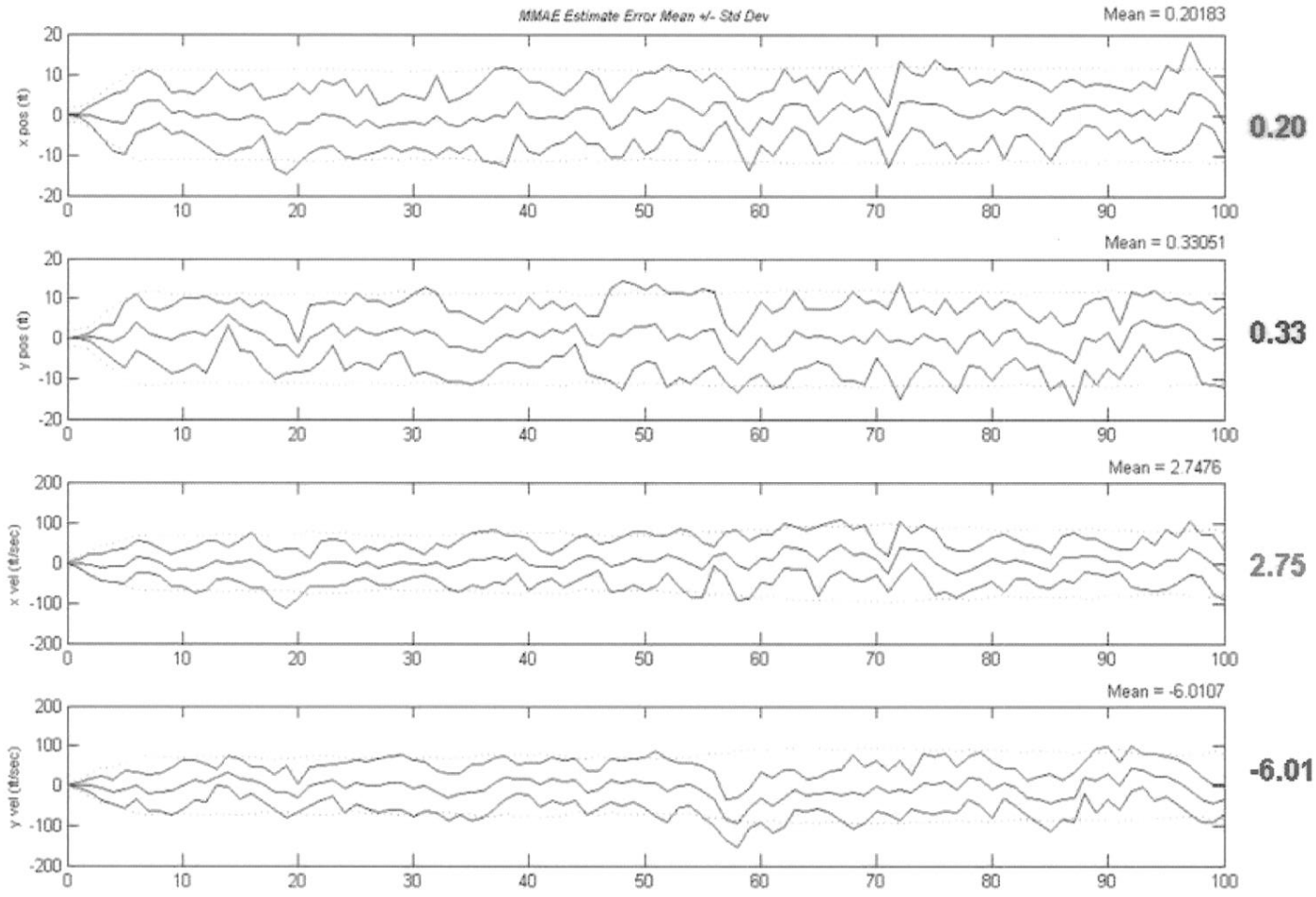


(a)

Figure 28: (a) Modal probability flow averaged over 10 Monte Carlo simulations

We designed the PCA filters for step maneuvers; however, the probability flow never fully transitioned to the other filters after the maneuver. This shows the "coarseness" of

our discretization of parameter space meaning there is no one filter which takes all the probability flow from the other filters during a maneuver.



(b)

Figure 28. (b) Overall MMAE RMS position and velocity error

4.2.4 MMAE Design Conclusions

The FOGMA with PCA Complement Filters and SOGMA filter design appears to work well (i.e., produces low position error) under the test conditions of the four test

cases. Incorporation of the SOGMA filter seemed to improve RMS position error results but never obtained the majority of the MMAE modal probability for any significant amount of time. Again, the previous section simply justifies this MMAE being a decent or reasonable MMAE design to use as a basis of comparison.

4.3 MMAE Results

The performance of the MWA has been tested against a sinusoidal function, an aircraft kinematic model (benign and harsh maneuvering), and an isotropic rocket model (benign and maneuvering). The above seven-filter MMAE was put to the test against the MWA for comparison keeping in mind that the MMAE cannot estimate the unknown control signal. Therefore, the ability to track the position and velocity was used as a basis of comparison between the MMAE and MWA.

4.3.1 One Dimensional Sinusoid Tracking

The first test case was the sinusoidal function which simulates an aircraft performing a "jinking" maneuver.

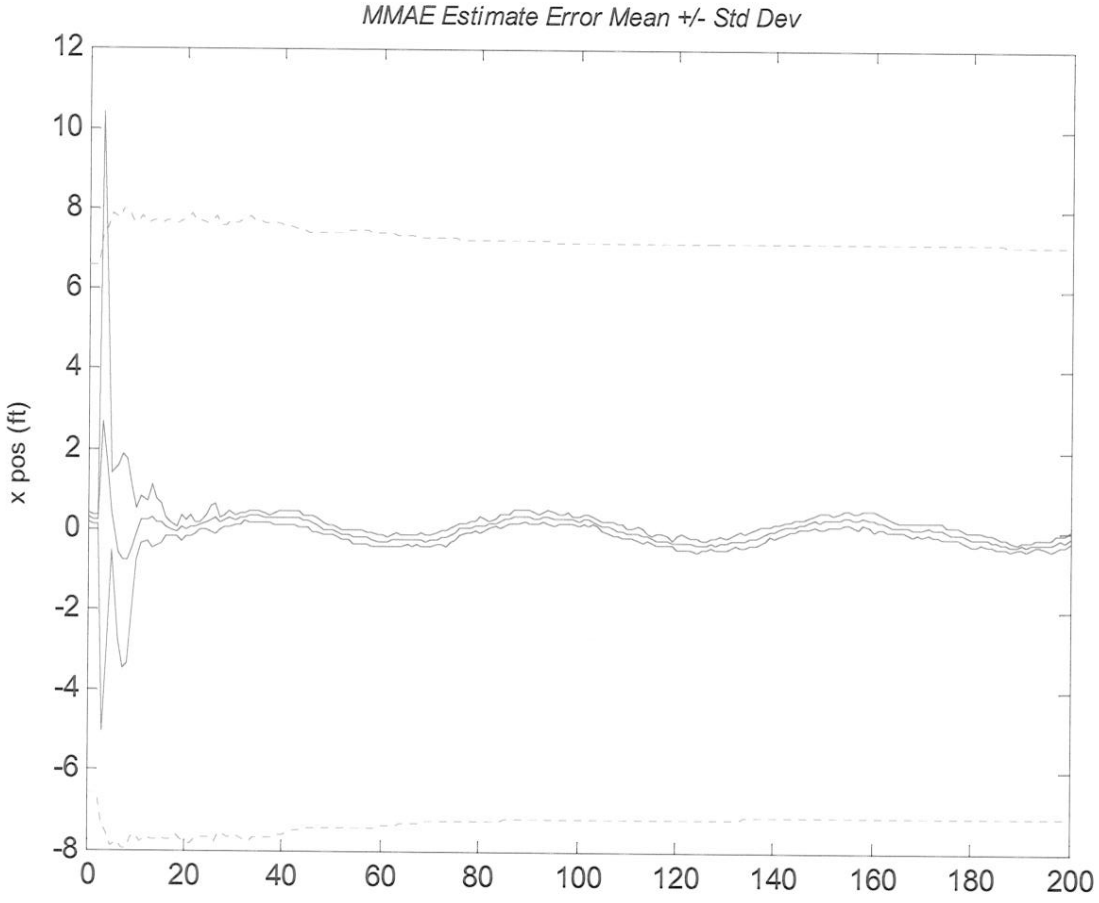


Figure 29: MMAE Estimate Error Mean +/- One Sigma (Monte Carlo evaluated) (10 Monte Carlo runs) ($dt = 0.1$)

RMS errors are satisfactorily small. Notice it takes on the order of 20 time steps or 2 seconds for best tracking performance.

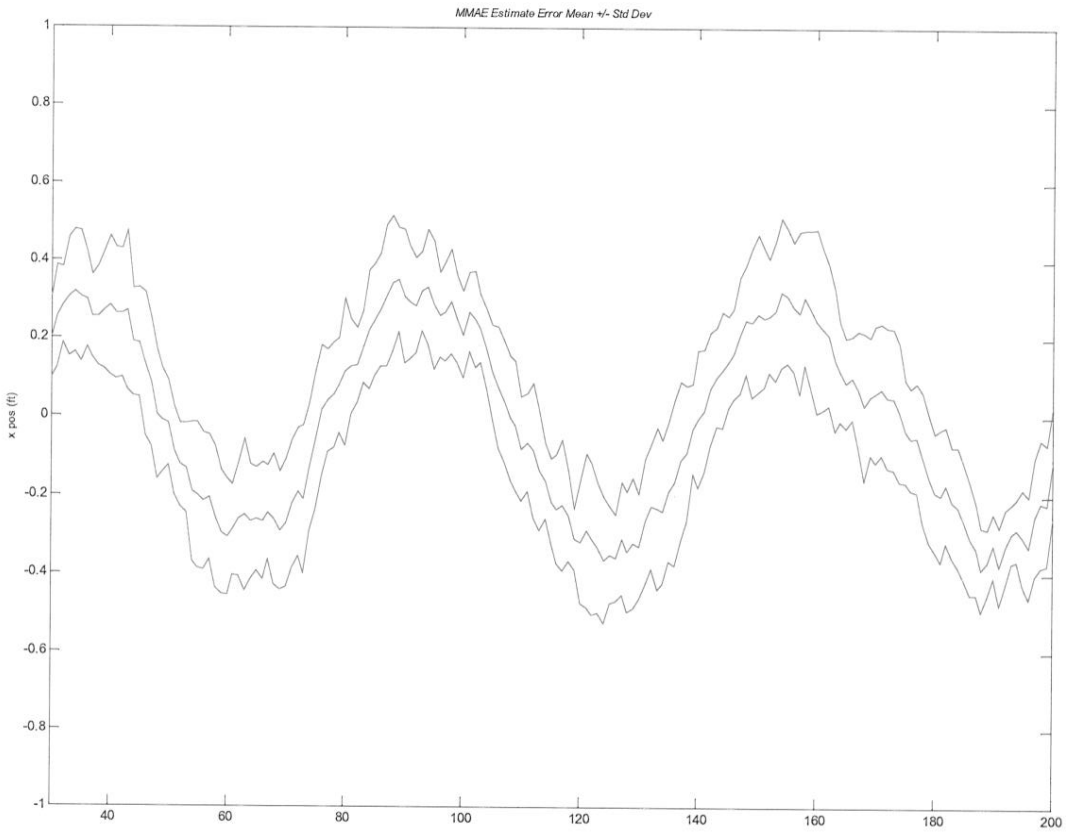


Figure 30: Zoomed in MMAE Estimate Error Mean +/- One Sigma (Monte Carlo evaluated) (10 Monte Carlo Runs)

Figure 30 presents a zoomed version of Figure 29 to resolve best tracking performance. The MMAE based filter tracks very well.

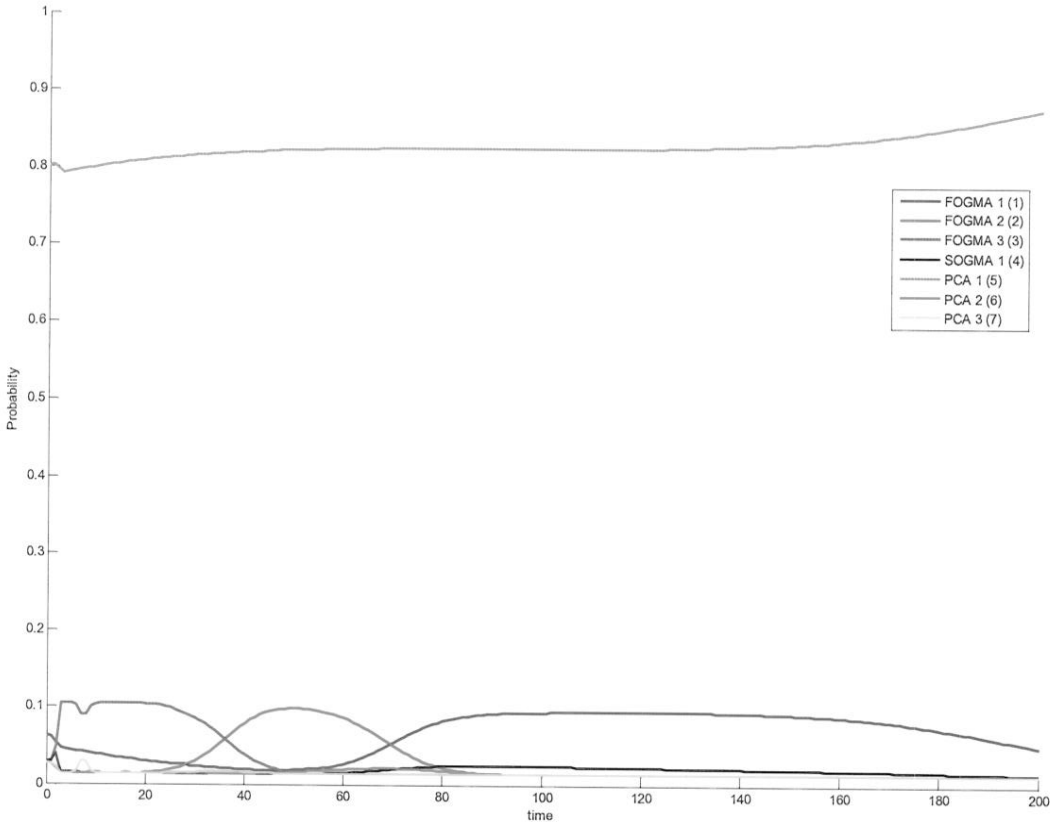


Figure 31: Sinusoidal Tracking. Probability Flow of MMAE

The PCA 1 filter assumes most of the probability and never "lets go". This is probably the result of the frequency of the sinusoid function ($1/2\pi$). Note the alternating probability flow between all three FOGMA models. This unfortunately implies that this MMAE is not necessarily the "right" MMAE to use for portraying the best that an MMAE can provide in performance for this particular problem.

4.3.2 Aircraft Kinematic Model

The MMAE filter is tested against the aircraft motion model using the same truth model parameters used for the MWA in Section 4.1.4.

Table 9: Parameters for MMAE Aircraft Kinematic Model Simulation

$V_{true} = 200\,\text{m/s}$
$\sigma_k = 40.7\,\text{m}$
$\Omega = 0.0982$ rad/sec

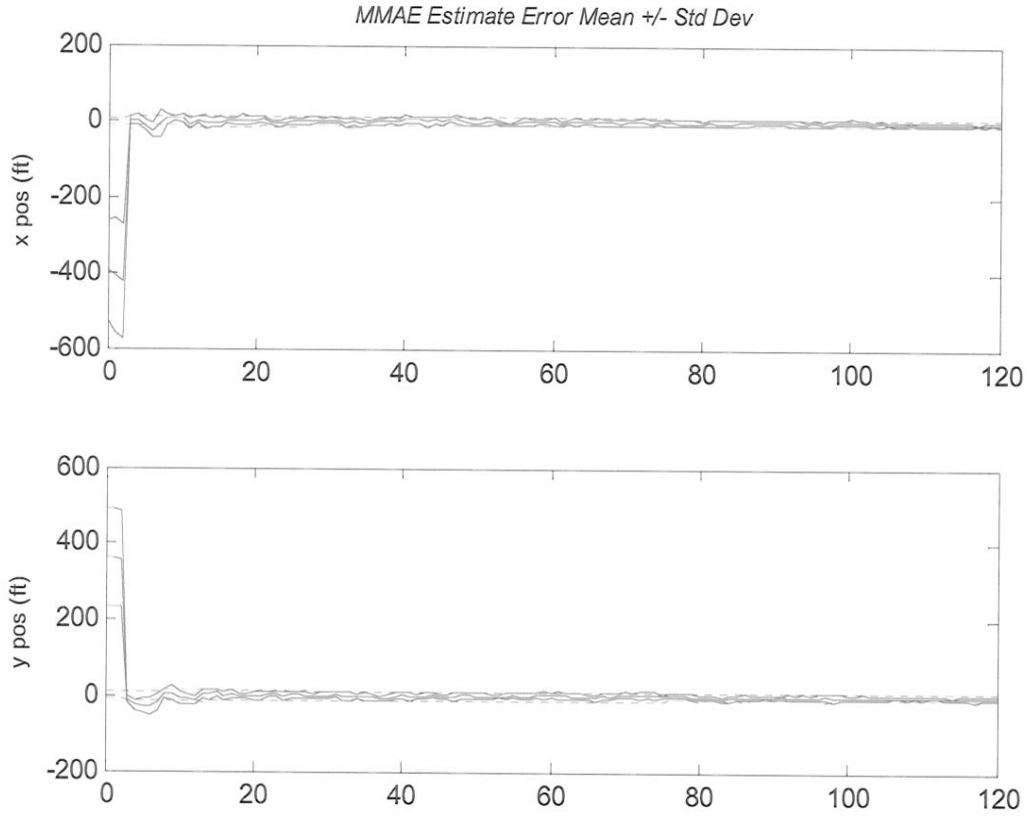


Figure 32: MMAE Estimate Error Mean +/- One Sigma (Monte Carlo evaluated) (10 Monte Carlo Runs) ($dt = 0.1$)

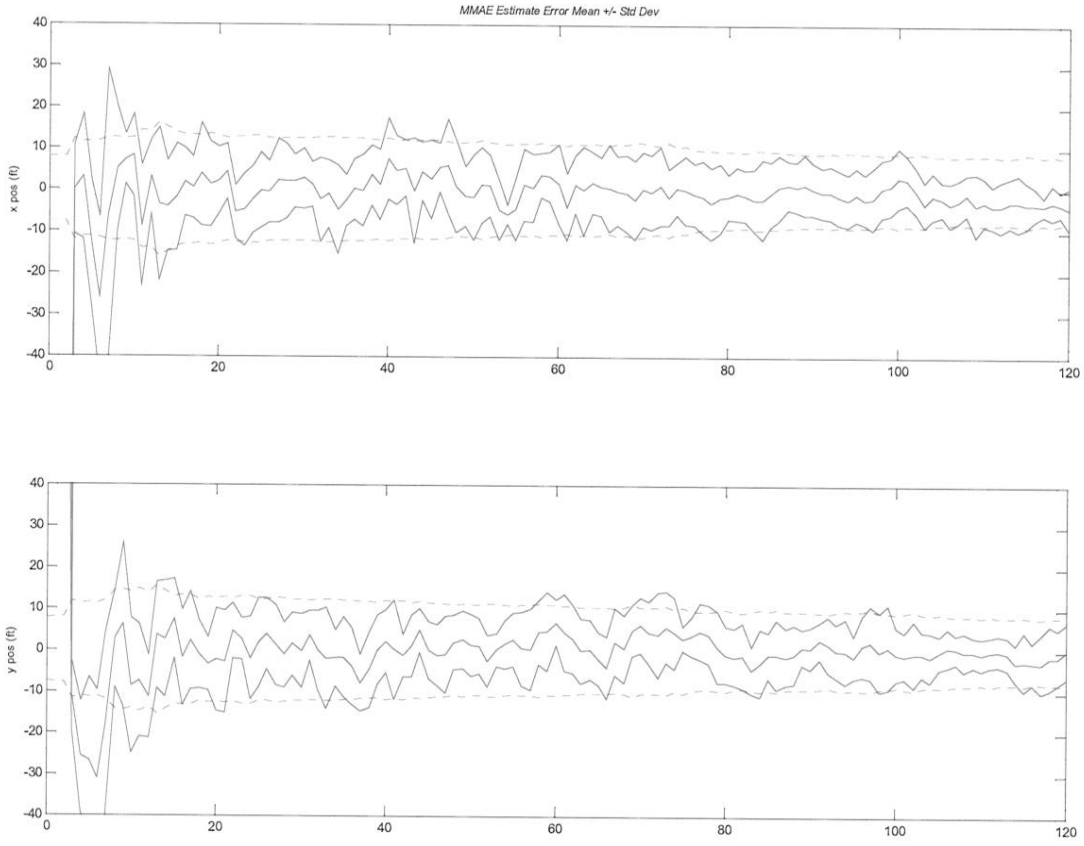


Figure 33: Zoomed in MMAE Estimate Error Mean +/- One Sigma (Monte Carlo evaluated)

The MMAE performs reasonably considering the intensity of measurement noise. RMS errors are not as good as those from the MWA. The MWA performed an order of magnitude better in precision.

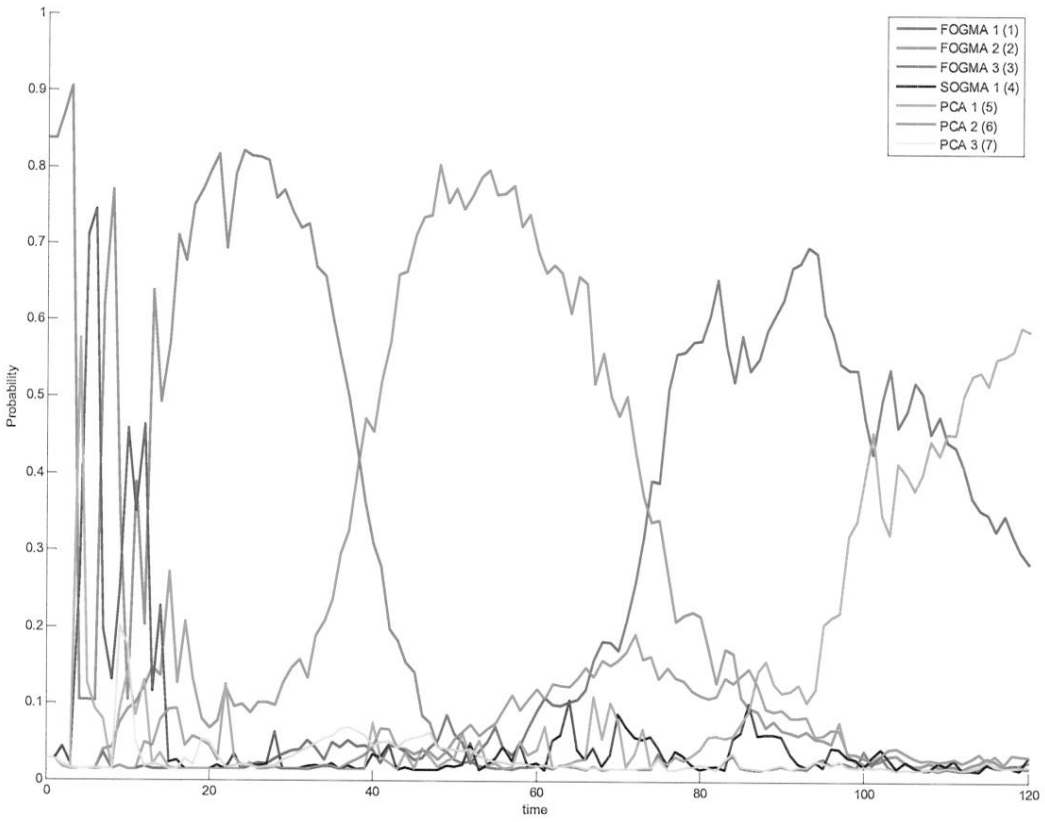


Figure 34: MMAE Aircraft Kinematic Model. Probability Flow of MMAE

The alternating probability flow between the three FOGMA models appears again. The PCA 1 filter is silent until towards the end of the simulation. The cycling between the FOGMA models may be indicative of the sinusoidal nature of the maneuver.

4.3.2.1 Maneuvering Aircraft Model

The MMAE is now tested against a maneuvering aircraft model.

Table 10. MMAE Parameters for Maneuvering Aircraft Simulation

$0 \le t < 40$	$40 \le t < 80$	$80 \le t < 120$	$120 \le t < 160$
$V_{true} = 200$ m/s	$V_{true} = 100$ m/s	$V_{true} = -300$ m/s	$V_{true} = 50$ m/s
$\Omega = 0.098$ rad/sec	$\Omega = 0.2$ rad/sec	$\Omega = 0.3$ rad/sec	$\Omega = 0.5$ rad/sec

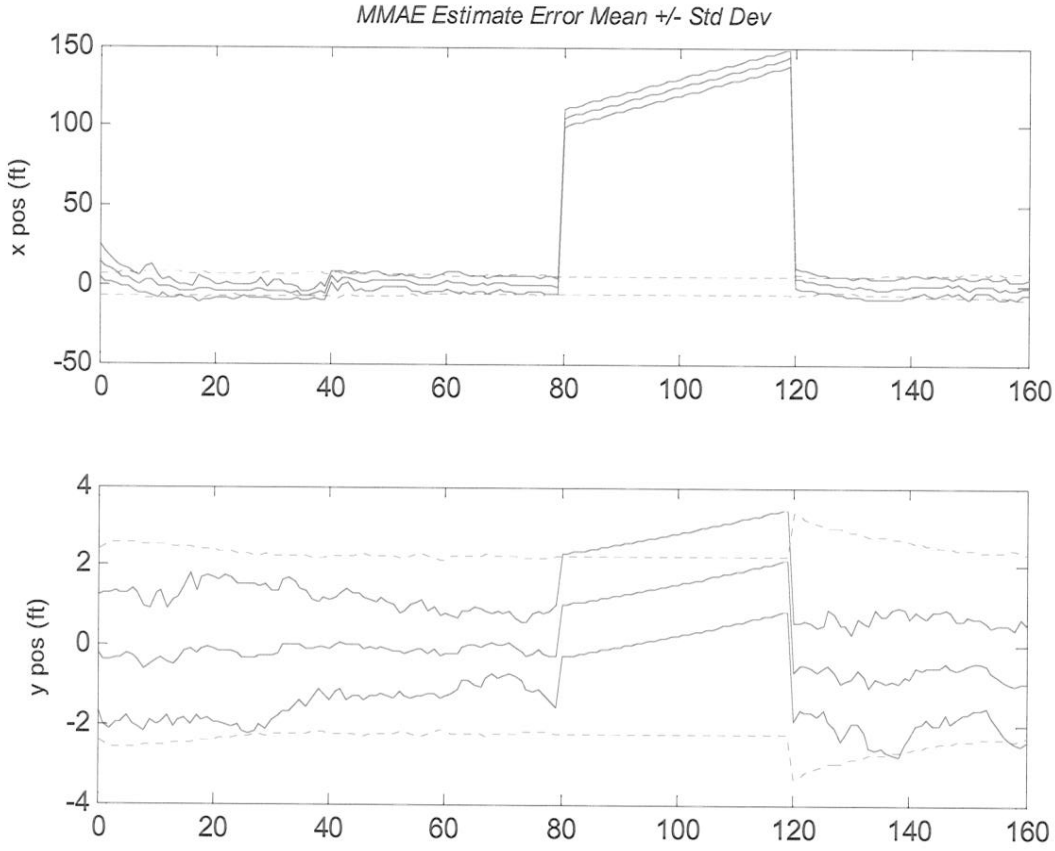


Figure 35: MMAE Estimate Error Mean +/- One Sigma (Monte Carlo evaluated) (10 Monte Carlo runs) (*dt =0.001*)

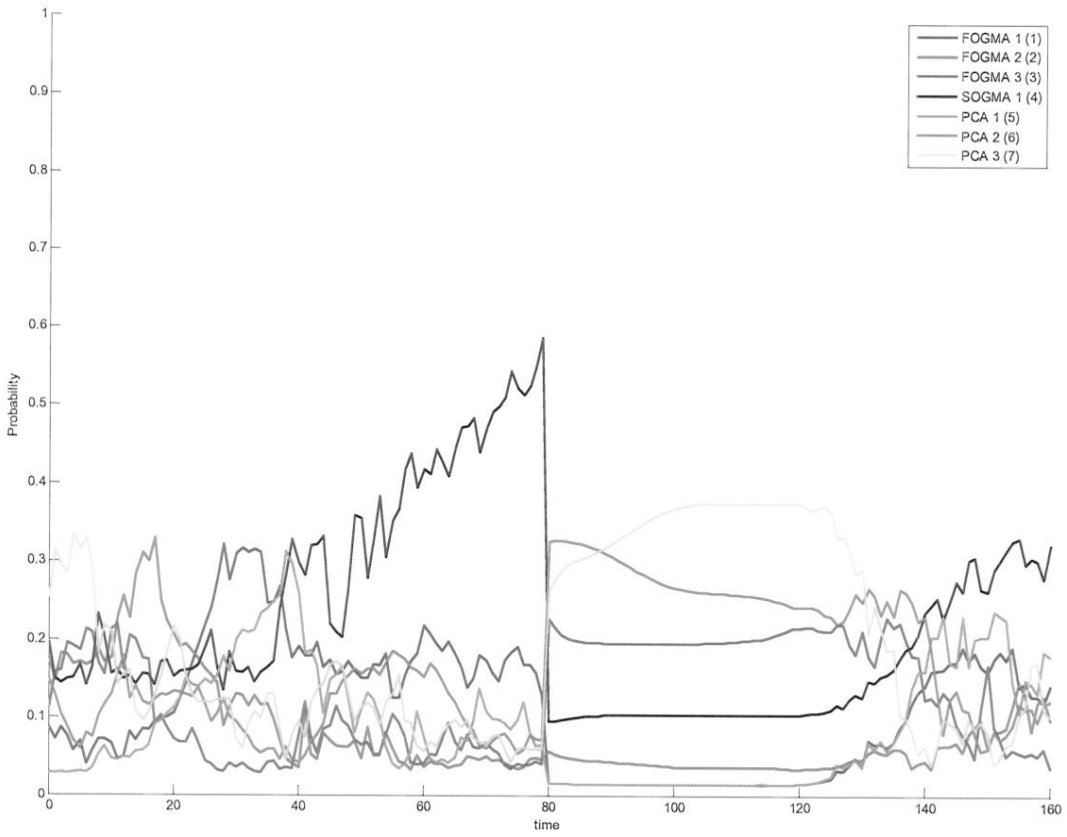


Figure 36: MMAE Maneuvering Aircraft Kinematic Model. Probability Flow of MMAE.

The MMAE based tracking filter had the most difficulty during the third maneuver. A major problem must exist in the MMAE and was regrettably unresolved. It tracked quite well during the other maneuvers and primarily equaled the error characteristics of the MWA. In addition, without the considerably small $dt = 0.001$, all filters in the MMAE diverged. Nonetheless, the MMAE produced good tracking performance for the maneuvering aircraft kinematic model.

4.3.3 Smooth Sinusoidal Isotropic Rocket Model

MMAE is now tested against the Smooth Sinusoidal Isotropic Rocket Model. The truth model parameters for the maneuvering target ($dt = 0.1$) are listed in Table 11.

Table 11: MMAE Parameters for Smooth Isotropic Rocket Model Inputs

$0 \le t < 100$	$100 \le t < 200$	$200 \le t < 300$	$300 \le t < 400$
$A = 5\sqrt{2}$	$A = 2\sqrt{2}$	$A = 4\sqrt{2}$	$A = 10\sqrt{2}$
$\omega = \frac{\pi}{4}$ rad/sec	$\omega = \frac{\pi}{2}$ rad/sec	$\omega = \frac{\pi}{12}$ rad/sec	$\omega = \frac{\pi}{6}$ rad/sec

It should be mentioned that the MMAE-based tracking filter could not maintain track utilizing the exact parameters from the MWA simulation. The amplitudes of the sinusoidal truth model were reduced an order of magnitude to prevent divergence of all filters. This further illustrates that despite the intention of a fair comparison between MWA and MMAE, there is strong reason to believe that this MMAE is flawed, so that a meaningful comparison has unfortunately not been provided.

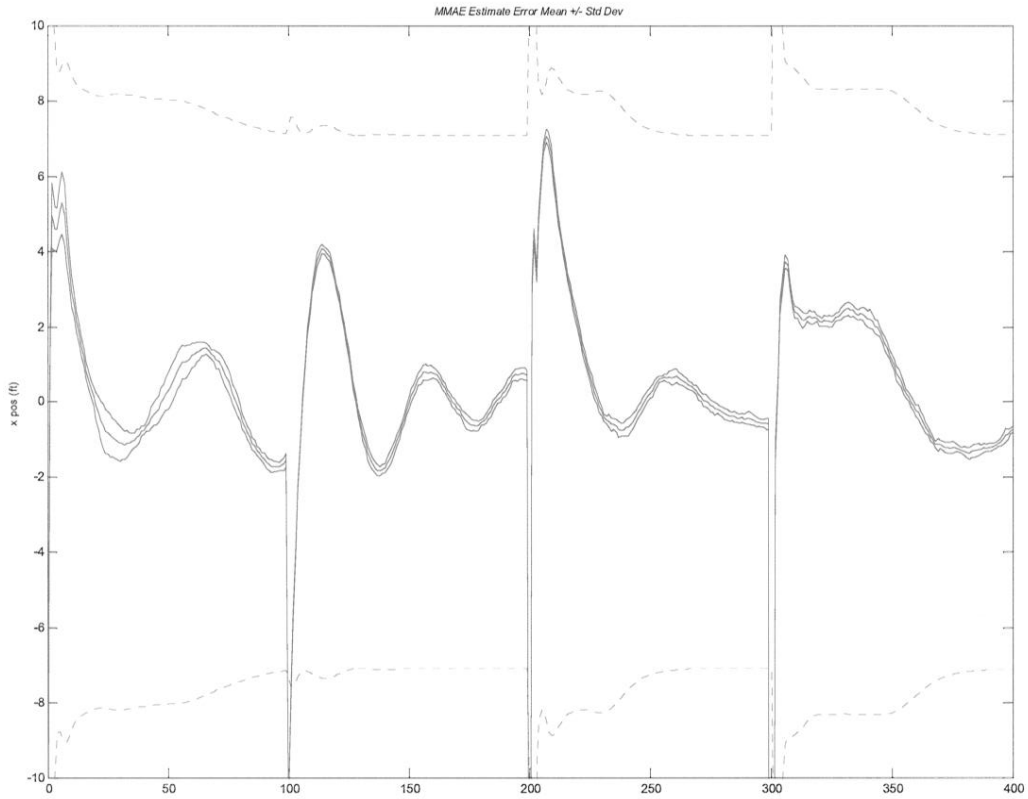


Figure 37: Mean MMAE Estimate Error +/- One Sigma (Monte Carlo evaluated)

(10 Monte Carlo runs) ($dt = 0.1$)

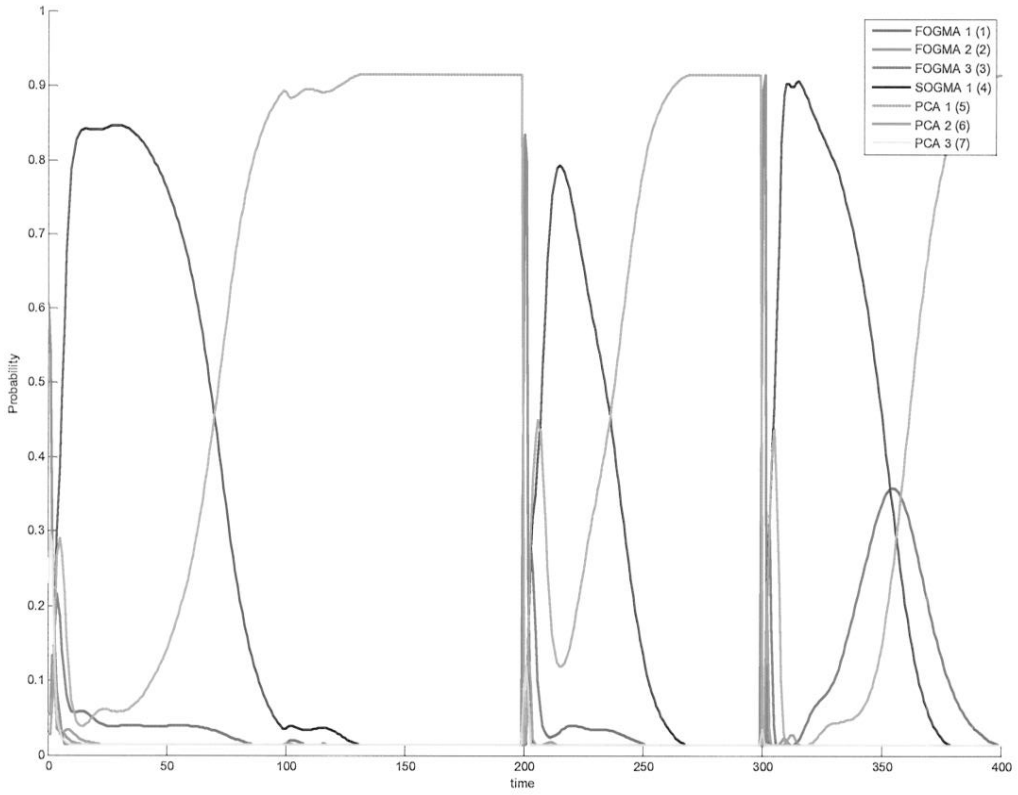


Figure 38: Probability Flow of MMAE

The SOGMA and PCA 1 filters took the majority of the probability flow.

4.4 Summary

In summary, comparing the MMAE-based filter was quite difficult. This comparison showed that tuning an MMAE-based filter is more complex than tuning our moving window algorithm. These results do not imply that an MMAE-based filter is incapable of providing excellent position and velocity tracks but that this author achieved better success using the moving window algorithm. We obtained good estimates of position and velocity as well as an estimate of the input.

V. Discussion

This chapter discusses the experimental results of this research. It will discuss insight from the calibration experiment, the MWA, the smoother used in the algorithm, and the MMAE comparison.

5.1. Calibration Experiment

The calibration experiment using the constant acceleration model provided great insight into estimation window length setting. An estimation window on the order of a second long and using thirty sample periods within each window produced good estimates. We recognize a longer β would improve KF performance, e.g., when increasing window length to 50 or 100 sample periods. The RMS estimation error reduction did not improve enough to warrant such an increase in computation time.

5.2 Moving Window Algorithm

The input-estimation-based Moving Window Algorithm (MWA) performed exceptionally well, independent of the complexity of the trajectory it was designed to track. Small estimation window lengths between 0.5 and 1 second worked very efficiently. These small intervals allowed estimation of the various trajectories with minimal RMS errors. In addition, the MWA has the additional attribute of being able to detect maneuver onset. When sliding over a sudden change in input level, it spikes. Indeed, the discontinuity in the control signal causes the spike, because the control parameter is supposed to be constant in the window. This destroys the window's approximation of the control as it "slides" over it. We designate the highest derivative, the control variable and since the derivative does not exist at sharp turns (corners) or at

the moment of a step discontinuity, a spike forms. This makes this tracking filter a good maneuver detector. Such detection ability allows you to detect points where the constant u model is not adequate. One should not try and use estimates based on a window with a jump discontinuity. Instead, one should process up to the discontinuity, but not beyond, and process separately from that point forward seeking a different constant value for u in the next window. Nonetheless, it took about 2 seconds to regain track.

Most importantly, the MWA successfully estimated the control signal. Keeping the control signal almost constant during the small estimation window interval contributed to this result. The fact that the control level's variance is set very large at the beginning of each window is instrumental in achieving successful results with minimal RMS errors when the control signal changes rapidly. It allowed the tracking filter to maintain track, despite the most complex of trajectories.

5.3 Smoothing Algorithm

The moving window estimator would not be as effective without the smoothed estimates of the dynamic states (excluding the control signal u) at window onset. One important issue concerning moving estimation intervals is determining the initial conditions for of each estimation window. Smoothing the state estimates for the next estimation interval's initial conditions while the Kalman Filter ran concurrently achieved the best results. In all cases, the smoothing algorithm provided excellent tracking performance and helped to estimate the control signal u.

5.4 MMAE vs. MWA

A comparison was attempted between the use of one sliding window Kalman Filter and a seven-filter MMAE filter bank. This task proved quite difficult for a fair

comparison. MMAE requires much tuning of many Kalman Filters and relies upon proper discretization of the parameter space. This can seem to be a long and involved process to achieve minimal RMS errors independent of the complexity of an aircraft's maneuver. Indeed, the complexity of MMAE design motivated the objective of this thesis. The MWA has only two significant parameters of concern: the estimation interval length and the Kalman Filter's sampling rate (time step). For very strong maneuvers, the MMAE had difficulty maintaining track and diverged quite often. This is strange and alarming for a well-designed MMAE. Basing the MMAE elemental filters on the augmented model may have alleviated such issues. The MWA RMS errors were either on par or exceeded the MMAE design. The moving window with smoother algorithm proved to outperform the MMAE for most of the simulations and estimated the parameters, that is, provided good target tracking with good precision. Finally, by construction, the input estimation-based tracking filter provides an estimate of the target's acceleration, recognizing the precision is not as good as position and velocity estimates. This is conducive to the employment of advanced Augmented Proportional Navigation missile guidance laws which require an estimate of the target's acceleration.

5.5 Summary

This chapter discussed the results of the calibration experiment and explained the benefits of the moving window and smoothing algorithms. It also discussed how well the MMAE-based filter performed against the MWA.

VI. Conclusions and Recommendations

The focus of this study was on tracking maneuvering targets using input estimation and utilizing one Kalman Filter. An attempt to compare the results with a seven-filter MMAE was made. The input signal's level u is considered a continuous variable and consequently the input estimation problem is posed purely as a parameter estimation problem. This application of the new tracking filter algorithm is not contingent on distinguishing maneuvering and non-maneuvering targets, and does not require the detection of maneuver onset. The filter automatically detects the onset of a maneuver. Furthermore, an estimate of the target's acceleration is also obtained (just as FOGMA, SOGMA, and PCA filters can all provide). This opens the door to the employment of advanced Augmented Proportional Navigation missile guidance laws, which require an estimate of the target's acceleration state. When the target dynamics and measurement equation are linear and input u is constant, then an unbiased estimate $\hat{u}$ of u and of the target's state is obtained, provided that an observability condition holds. It is shown that the critical observability condition holds for kinematic target motion models of interest.

The MWA provides a source of maneuver detection and subsequently takes at most 2 seconds to regain track independent of maneuver initiated. These discontinuities can be used to make the algorithm run more efficiently. The spikes demonstrate where our constant u model is not adequate and can be used to halt estimates based on a window that includes this point of discontinuity. One could process up to that point in time using

some metric (residual monitoring), but not beyond, then process separately from the point in time forward seeking a different constant value through the estimator.

As with all tracking algorithms, computation time and numeric stability are very important. Parameter estimation traditionally requires much computation. A recommendation that would provide better numeric stability is using U-D Factorization. Moreover, in the smoothing algorithm, using Fraser's form of the fixed-point smoother may speed up computation. It eliminates the need for covariance matrix inversion and, instead, uses the measurement noise matrix inversion. If the number of measurements is substantially less than the number of states, this may reap big benefits in computational loading.

Bibliography

1. Bar-Shalom, Y and Xiao-Rong Li. *Estimation and Tracking: Principles, Techniques and Software.* Artech House, Norwood, MA, 1993, pp. 426-431.

2. Blackman, S., and Popoli R., *Design and Analysis of Modern Tracking Systems.* Norwood, Massachusetts, 1999, pp. 360-369.

3. Maybeck, P.S., *Stochastic Models, Estimation, and Control Volume 1.* Department of Electrical Engineering, AFIT, Wright Patterson Air Force Base Dayton, Ohio, 1982, pp. 172-174.

4. Maybeck, P.S., *Stochastic Models, Estimation, and Control Volume 2.* Department of Electrical Engineering, AFIT, Wright Patterson Air Force Base Dayton, Ohio, 1982, pp. 15-17, 129-136.

5. Rogers, S., *Alpha-beta filter with correlated measurement noise.* IEEE Transactions on Aerospace and Electronic Systems, 1987. AES-23: pp. 592-594.

6. Singer, R.A., *Estimating Optimal Tracking Filter Performance for Manned Maneuvering Targets.* IEEE Transactions on Aerospace and Electronic Systems, 1970. AES-6(4): pp. 473-483.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
13-06-2006	**Master's Thesis**	Aug 2004-Jun 2006

4. TITLE AND SUBTITLE

New Tracking Filter Algorithm Using Input Parameter Estimation

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Broussard, Corey M., Captain, USAF

5d. PROJECT NUMBER
If funded, enter ENR #

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GA/ENG/06-01

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

13. SUPPLEMENTARY NOTES

14. ABSTRACT

A new method for the design of tracking filters for maneuvering targets, based on kinematic models and input signals estimation, is developed. The input signal's level, u, is considered a continuous variable and consequently the input estimation problem is posed as a purely parameter estimation problem. Moreover, the application of the new tracking filter algorithm is not contingent on distinguishing maneuvering and non-maneuvering targets, and does not require the detection of maneuver onset. The filter will automatically detect the onset of a maneuver. Furthermore, an estimate of the target's acceleration is also obtained with reasonable precision. This opens the door to the employment of advanced Augmented Proportional Navigation Missile guidance laws, which require an estimate of the target's state acceleration, recognizing the precision is not as good as that for position and velocity estimates. When the target dynamics and measurement are linear and input u is constant, then an unbiased estimate of $\hat{u}$ of u and of the target state is obtained, provided that an observability condition holds. It is shown that the critical observability condition holds for kinematic target motion models of interest.

15. SUBJECT TERMS

Tracking filters, input estimation

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
REPORT	**ABSTRACT**	**c. THIS PAGE**			**Meir Pachter, PhD (ENG)**
U	**U**	**U**	**UU**	97	**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-3636 ext 7247; email: meir.pachter@afit.edu